

Documented Code For glossaries v4.50

Nicola L.C. Talbot

Dickimaw Books

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

2022-10-14

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 . .	106
2.10	Loading files containing glossary entries	112
2.11	Using glossary entries in the text	112
2.11.1	Links to glossary entries	118
2.11.2	Displaying entry details without adding information to the glossary	163
2.12	Adding an entry to the glossary without generating text	175
2.13	Creating associated files	176
2.14	Writing information to associated files	198
2.15	Glossary Entry Cross-References	208
2.16	Displaying the glossary	210
2.17	Acronyms	238
2.18	Predefined acronym styles	243
2.19	Predefined Glossary Styles	261
2.20	Debugging Commands	262
2.21	Compatibility with version 2.07 and below	268

3	Glossary Styles	268
3.1	Glossary hyper-navigation definitions (glossary-hypernav.sty) . . .	268
3.2	In-line Style (glossary-inline.sty)	271
3.3	List Style (glossary-list.sty)	273
3.4	Glossary Styles using longtable (glossary-long.sty)	277
3.5	Glossary Styles using longtable and booktabs (glossary-longbooktabs.sty)	284
3.6	Glossary Styles using longtable (glossary-longragged.sty)	289
3.7	Glossary Styles using multicol (glossary-mcols.sty)	294
3.8	Glossary Styles using supertabular environment (glossary-super.sty)	300
3.9	Glossary Styles using supertabular environment (glossary-superragged.sty)	307
3.10	Tree Styles (glossary-tree.sty)	314
4	Multi-Lingual Support	323
4.1	Babel Captions (glossaries-babel.sty)	323
4.2	Polyglossia Captions (glossaries-polyglossia.sty)	324
5	Prefix Support (glossaries-prefix Code)	325
6	Accessibility Support (glossaries-accsupp Code)	332
6.1	Defining Replacement Text	334
6.2	Accessing Replacement Text	338
6.3	Displaying the Glossary	358
6.4	Acronyms	360
6.5	Debugging Commands	370
7	Backwards Compatibility	371
7.1	glossaries-compatible-207	371
7.2	glossaries-compatible-307	371
8	Rollback	372
8.1	Rollback v4.46 (glossaries-2020-03-19.sty)	372
8.2	Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)	535
8.3	Rollback v4.46 (glossaries-babel-2020-03-19.sty)	570
8.4	Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)	571
8.5	Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)	576
8.6	Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)	587
8.7	Rollback v4.46 (glossaries-prefix-2020-03-19.sty)	587
8.8	Rollback v4.46 (glossary-hypernav-2020-03-19.sty)	592
8.9	Rollback v4.46 (glossary-inline-2020-03-19.sty)	593
8.10	Rollback v4.46 (glossary-list-2020-03-19.sty)	594
8.11	Rollback v4.46 (glossary-list-2020-03-19.sty)	596
8.12	Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)	599
8.13	Rollback v4.46 (glossary-longragged-2020-03-19.sty)	602
8.14	Rollback v4.46 (glossary-mcols-2020-03-19.sty)	605
8.15	Rollback v4.46 (glossary-super-2020-03-19.sty)	609
8.16	Rollback v4.46 (glossary-superragged-2020-03-19.sty)	613

8.17	Rollback v4.46 (glossary-tree-2020-03-19.sty)	616
8.18	Rollback v4.49 (glossaries-2021-11-01.sty)	622
8.19	Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	788
8.20	Rollback v4.49 (glossaries-babel-2021-11-01.sty)	823
8.21	Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	824
8.22	Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	830
8.23	Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	840
8.24	Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	840
8.25	Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	845
8.26	Rollback v4.49 (glossary-inline-2021-11-01.sty)	846
8.27	Rollback v4.49 (glossary-list-2021-11-01.sty)	848
8.28	Rollback v4.49 (glossary-long-2021-11-01.sty)	850
8.29	Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	853
8.30	Rollback v4.49 (glossary-longragged-2021-11-01.sty)	856
8.31	Rollback v4.49 (glossary-mcols-2021-11-01.sty)	859
8.32	Rollback v4.49 (glossary-super-2021-11-01.sty)	862
8.33	Rollback v4.49 (glossary-superragged-2021-11-01.sty)	866
8.34	Rollback v4.49 (glossary-tree-2021-11-01.sty)	870

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** v4.50: L^AT_EX2e Package to Assist Generating Glossaries”.

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

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

INSTALL Installation instructions.

CHANGES Change log.

README.md Package summary.

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

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

2 Main Package Code (glossaries.sty)

2.1 Package Definition

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossaries}[2022/10/14 v4.50 (NLCT)]
```

Required packages:

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

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

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

```
\ExplSyntaxOn
```

`\glslowercase` Provided to make it easier for user to use `LATEX3` lowercase command without switching to `exp3` syntax. For example, to redefine `\acronymfont` to automatically convert to lowercase.

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

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

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

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

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

Automatically identify exclusions, blockers and mappings.

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

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

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

}
```

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

```
\RequirePackage{textcase}
```

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

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

Provide support for exclusions with `\MFUsentencecase`.

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

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

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

```

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

```

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

```

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

```

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

```

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

```

`\glsshowtargetouter`

```

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

```

`\glsshowtargetsymbolsymbol`

```

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

```

`\glsshowtargetfont`

```

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

```

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

```

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

```

`\glsdoshowtarget` Make it easier to customise.

```

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

```

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

```

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

```


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

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

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

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

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

`\@gls@see@noindex`

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

`seenindex`

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

```

\fi
}

```

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

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

```

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

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

```

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

```

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

```

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

```

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

```

Determine whether or not to use numbered sections.

\@@glossarysecstar

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

```

\@@glossaryseclabel

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

```

\glsautoprefix Prefix to add before label if automatically generated:

```
\newcommand*\glsautoprefix{}

```

numberedsection

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

`\@gls@loadlong`

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

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

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

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

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

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

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

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

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

`\@gls@loadlist`

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

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

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

`\@gls@loadtree`

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

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

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

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

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

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

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

nopostdot Boolean option to suppress post description dot

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

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

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

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

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

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

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

`\define@glossaryentrycounter`

```
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
    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\theH@glossaryentry{\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]{}
\glentrycounterfalse
```

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

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

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

`\define@glossarysubentrycounter`

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

`\glsprestandardsort`

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

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

`\@glo@check@sortallowed`

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

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

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

Store entry information when it's defined.

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

No count register required for standard sort.

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

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

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

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

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

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

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

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

Set standard sort as the default:

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

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

```
\newcommand*\glssortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
```


`\@gls@setupsort@def` Set up the macros for order of definition sorting.

```

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

```

`\@gls@setupsort@use` Set up the macros for order of use sorting.

```

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

```

Get the parent, if one exists

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

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

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

Set index information for this entry

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

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

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

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

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

Don't store entry index information.

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

No count register required for standard sort.

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

Don't modify sort value.

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

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

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

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

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

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

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

Don't store entry index information.

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

No count register required for standard sort.

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

Clear sort value.

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

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

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

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

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

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

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

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

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

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

`\glsdefaulttype`

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

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

`\acronymtype`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

    }%
  }%
}

```

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

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

`\@addtoacronymlists`

```

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

```

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

```

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

```

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

`\glsIfListOfAcronyms`

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

```

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

```

Internal command requires label and list to be expanded:

```

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

```

Not found

```

    #4%
  \else

```

Found

```

    #3%
  \fi
}

```

`\if@glsisacronymlist` Convenient boolean.

```

\newif\if@glsisacronymlist

```

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

```

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

```

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

```

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

```

`acronymlists`

```

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

```

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

`\glscounter`

```

\newcommand{\glscounter}{page}

```

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

```

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

```

`\@gls@nohyperlist`

```

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

```

`\GlsDeclareNoHyperList`

```

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

```

nohypertypes

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

\GlossariesWarning Prints a warning message.

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

\GlossariesWarningNoLine Prints a warning message without the line number.

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

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

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

nowarn Define package option to suppress warnings

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

\@gls@missinglang@warn Missing language warning.

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

nolangwarn Suppress warning if language support not found.

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

```

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

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

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

```

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

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

`\glssetexpandfield`

`\glssetexpandfield{<field>}`

Sets field to always expand.

```

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

```

`\glssetnoexpandfield`

`\glssetnoexpandfield{<field>}`

Sets field to never expand.

```

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

```

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

`\glssetexpandfield{type}`

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

`\glssetnoexpandfield{desc}`


```

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

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

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

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

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

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

Sanitizing stuff:

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

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

\@gls@nosanitizesort
\newcommand*\@gls@nosanitizesort{%

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

```

\@@gls@noidx@nosanitizesort

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

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

```

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

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

```

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

```

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

```

\@gls@removedoption

```

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

```

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

```

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

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

```

```

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

```

sanitizesort

```

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

```

Default setting:

```

\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%

```

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

```

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

```

```

    deprecated. Use sanitizesort instead}%
}

```

sanitize

```

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

```

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

```

\newif\ifglstranslate

```

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

`\@gls@usetranslator`

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Set the default value:

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

`\@gls@requested@languages`

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

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

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

indexonlyfirst Set whether to only index on first use.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

shortcuts Define acronym shortcuts.

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

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

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

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

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

order

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

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

```
\newif\ifglxindy
```


The default is `makeindex`:

```
\glsxindyfalse
```

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

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

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

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

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

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

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

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

Define key to set the language

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

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

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

Define a key to set the code page.

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

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

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

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

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

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

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

`\@domakeglossaries`

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

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

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

`restoremakegloss` Cancel the effect of `disablemakegloss`.

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

```

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

```

\do@write@glslabels

```

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

```

\@do@write@glslabels

```

\newcommand*{\@do@write@glslabels}{%
\newwrite\@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
}

```

\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}

```

`\ifglautomake`

```

\newif\ifglautomake

```

`\gls@automake@nr`

```

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

```

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

```

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

```

`\gls@automake@option`

```

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

```

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

```

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

```

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

```

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

```

```

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

```

`\@gls@doautomake`

```

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

```

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

```

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

```

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

```

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

```

```

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

```

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

```

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

```

Set default:

```

\glssavewritesfalse
\let\glswritefiles\@empty

```

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

```

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

```

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

```

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

```

\gls@original@makeglossary Store the original definition.

```

\let\gls@original@makeglossary\makeglossary

```

\gls@original@glossary Store the original definition.

```

\let\gls@original@glossary\glossary

```

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

`\@gls@override@glossary`

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

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

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

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

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

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

```

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

symbols Create a “symbols” glossary type
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}

Default is not to define the symbols glossary:
\newcommand*{\@gls@do@symbolsdef}{%

\@gls@symbolsdef

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

Define hook to set the toc title when translator is in use.
\newcommand*{\gls@tr@set@symbols@toctitle}{%
  \translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%

numbers Create a “symbols” glossary type
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}

Default is not to define the numbers glossary:
\newcommand*{\@gls@do@numbersdef}{%

\@gls@numbersdef

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

Define hook to set the toc title when translator is in use.
\newcommand*{\gls@tr@set@numbers@toctitle}{%
  \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
}%
}%

index Create an “index” glossary type
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef

```



```

\fi
}

```

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

```

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

```

Default is not to define index glossary:

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Now process options passed to the package:

```
\ProcessOptionsX
```

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

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

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

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

Now define `\setupglossaries`:

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

```

\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

2.3 Predefined Text

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

Main glossary title:

```

\glossaryname
\providecommand*\glossaryname{Glossary}

```

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

```

\acronymname
\providecommand*\acronymname{Acronyms}

```

```

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

```

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

```

\entryname
\providecommand*\entryname{Notation}

```

```

\descriptionname
\providecommand*\descriptionname{Description}

```

```

\symbolname
\providecommand*\symbolname{Symbol}

```

```

\pagelistname
\providecommand*\pagelistname{Page List}

```

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

```

\glssymbolsgroupname
\providecommand*\glssymbolsgroupname{Symbols}

```

```

\glnumbersgroupname
\providecommand*\glnumbersgroupname{Numbers}

```

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

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

`\glsacrpluralsuffix` Default plural suffix for acronyms

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

`\glsupacrpluralsuffix`

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

`\seename`

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

`\andname`

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

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

`\RequireGlossariesLang`

```
\newcommand*{\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}%
}
```

`\ProvidesGlossariesLang`

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

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

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

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

`\ifglstranslate`

Load tracklang:

```
\RequirePackage{tracklang}
```

Track any languages explicitly requested via the languages option.

```
\ifdefined\@gls@requested@languages{%
  {%
    \for\@gls@thislang:=\@gls@requested@languages\do{%
      \ifcsdef{@tracklang@add@\@gls@thislang}%
        {\csuse{@tracklang@add@\@gls@thislang}}%
        {\expandafter\TrackLanguageTag{\@gls@thislang}}%
    }%
  }%
}
```

Load translator if required.

```
\@gls@usetranslator
```

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

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

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

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

Check for tracked languages

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

```

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

```

if using translator use translator interface.

```

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

```

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

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

`\@nopostdesc` Suppress next description terminator.

```

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

```

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

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

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

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

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

```
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
  Just in case \istfilename has been modified.
  \ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
```

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

```
\@onlypremakeg\setStyleFile
```

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

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

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

The `makeglossaries` Perl script picks up this name from the auxiliary file. If the name ends with `.xdy` it calls `xindy` otherwise it calls `makeindex`. Since its not required by 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 L^AT_EX use a full stop as the compositor, which is why I have used it as the default.) If `xindy` is used `\glscompositor` only affects the `arabic-page-numbers` location class.

`\@glsAlphaCompositor` This is only used by `xindy`. It specifies the compositor to use when location numbers are in the form $\langle letter \rangle \langle compositor \rangle \langle number \rangle$. 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@suffixF` Suffix to use for a two page list. This overrides the separator and the closing page number if set to something other than an empty macro.

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

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

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

Only has an effect when used before `\makeglossaries`

`\@onlypremakeg\glsSetSuffixF`

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

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

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

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

`\glsnumberformat` The command `\glsnumberformat` indicates the default format for the page numbers in the glossary. (Note that this is not the same as `\glossaryentrynumbers`,

but applies to individual numbers or groups of numbers within an entry’s associated number list.) If hyperlinks are defined, it will use `\glshypernumber`, otherwise it will simply display its argument “as is”.

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

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

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

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

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

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

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

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

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

`\setglossarypreamble`

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

Code provided by Michael Pock.

```
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
```

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

```
%\renewcommand{\glossarypostamble}{For a complete list of terms
%see \cite{blah}\gdef@glossarypreamble{}}
%
```

`\glossarypostamble`

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

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

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

  \glsglossarymark{\glossarytoctitle}%
}
```

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

```
\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
```

```

\ifglsucmark
  \markboth{\memUHead{#1}}{\memUHead{#1}}%
\else
  \markboth{#1}{#1}%
\fi
}
}%
{%
\newcommand{\glsglossarymark}[1]{%
  \ifglsucmark
    \mkboth{\glssuppercase{#1}}{\glssuppercase{#1}}%
  \else
    \mkboth{#1}{#1}%
  \fi
}
}
}

```

`\glossarymark` Provided for backward compatibility:

```

\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \mkboth{\glssuppercase{#1}}{\glssuppercase{#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 `glstoc` flag set. If it is set, `\@gls@toc` will add a line to the `.toc` file, otherwise it will do nothing. (The first argument to `\@gls@toc` is the title for the table of contents, the second argument is the sectioning type.)

```

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

```

2.4 Xindy

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

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

```

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

```

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

```

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

```

`\@xdyattributes` Define list of attributes (`\string` is used in case the double quote character has been made active)

```

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

```

`\@xdyattributelist` Comma-separated list of attributes.

```

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

```

`\@xdylocref` Define list of markup location references.

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

`\@gls@ifinlist`

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

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

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

Check if already in list before adding.

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

Only has an effect before `\writeist`:

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

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

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

```

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

```

`\GlsAddXdyAttribute` Adds an attribute.

```
\ifglxsindy
```

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

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

Add to xindy attribute list

```

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

```

Add to xindy markup location.

```

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

```

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

```

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

```

High-level command:

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

Add to comma-separated attribute list

```

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

```

First add the attribute without the counter name:

```

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

```

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

```

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

```


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

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

Only has an effect before `\writeist`:

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

`\glscaddpredefinedattributes` Add known attributes for all defined counters

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

\GlsAddXdyAttribute{glignore}
}
\else
\let\@glscaddpredefinedattributes\relax
\fi
```

`\@xdyuseralphabets` List of additional alphabets

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

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

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

```

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

```

This code is only required for xindy:

```
\ifglsxindy
```

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

```

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

```

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

`\Lclass@Appendix-page-numbers` Appendix style locations (e.g. A-1, A-2, ..., B-1, B-2, ...). The separator is given by `\@glsAlphacompositor`.

```

\expandafter\def\csname @gls@xdy\Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \@glsAlphacompositor\string"
  \string"arabic-numbers\string"%
}

```

`\Lclass@arabic-section-numbers` Section number style locations (e.g. 1.1, 1.2, ...). The compositor is given by `\glscompositor`.

```

\expandafter\def\csname @gls@xdy\Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%

```

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

```

\def\@xdyuserlocationdefs{}

```

`\@xdyuserlocationnames` List of additional user location names

```

\def\@xdyuserlocationnames{}

```

End of xindy-only block:

```

\fi

```

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

```

\ifglxindy
  \newcommand\@xdycrossrefhook{}
\fi

```

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

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

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

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

Only has an effect before \writeist:

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

\@xdylocationclassorder Define location class order

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

Change the location order.

`\GlsSetXdyLocationClassOrder`

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

`\@xdysortrules` Define sort rules

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

`\GlsAddSortRule` Add a sort rule

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

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

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

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

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

`\GlsSetXdyStyles` Reset the list of required styles

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

```

```

\glsnoxywarning\GlsSetXdyStyles}
\fi

```

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

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

```

\def\@xdylanguage#1#2{}

```

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

```

\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]{%

```

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

`\@xdylettergroups` Store letter group definitions.

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

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

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

2.5 Loops and conditionals

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

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

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

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

`\forallacronyms`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{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 **\glentrytext** will no longer be usable in sectioning, caption etc commands. If the user really wants to be able to

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

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

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

`\glsdetoklabel`

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

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

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

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

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

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

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

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

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

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

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

`\glsdoifexists`

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

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

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

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

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

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

`\glsdoifexistsorwarn`

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

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

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

`\glsdoifexistsordo`

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

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

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

`\doifglossarynoexistsordo`

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

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

}

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

`\ifglshaschildren`

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

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

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

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

`\ifglshasparent`

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

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

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

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

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

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

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

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

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

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

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

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

`\ifgl$fieldvoid`

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

```
\newcommand{\ifglshasfieldvoid}[4]{\ifcvoid{glo@\glsdetoklabel{#2}@#1}{#3}{#4}}
```

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

\ifglshasfield

```
\newrobustcmd*{\ifglshasfield}[4]{%  
  \glsdoifexists{#2}%  
  {%  
    \letcs{\@glo@thisvalue}{glo@\glsdetoklabel{#2}@#1}%
```

First check supplied field label is defined.

```
\ifdef\@glo@thisvalue  
{%
```

Is defined, so now check if empty.

```
\ifdefempty\@glo@thisvalue  
{%
```

Is empty, so doesn't have field set.

```
  #4%  
}%  
{%
```

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

```
\ifdequal\@glo@thisvalue\@gls@default@value  
{%
```

Value is set to the default value.

```
  #4%  
}%  
{%
```

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

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

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

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

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

```
\ifdef\@gls@thisfield  
{%
```

Is defined, so now check if empty.

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

Is empty so field hasn't been set.

```
#4%
}%
{%
```

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

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

Value is set to the default value.

```
#4%
}%
{%
```

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

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

Not defined.

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

\glscurrentfieldvalue

```
\newcommand*{\glscurrentfieldvalue}{}%
```

2.6 Defining new glossaries

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

\@glo@types

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

```

\@gls@provide@newglossary If the user removes the glossary package from their document, ensure the next
run doesn't throw a load of undefined control sequence errors when the aux file
is parsed.
    \newcommand*\@gls@provide@newglossary{%
        \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}}%
Only need to do this once.
    \let\@gls@provide@newglossary\relax
    }

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

\defglsentryfmt Allow different glossaries to have different display styles.
    \newcommand*\defglsentryfmt[2][\glsdefaulttype]{%
        \protected@edef\@glo@tmp{#1}%
        \ifx\@glo@tmp\@gls@defaulttype
Argument was \glsdefaulttype but it hasn't expanded, which probably means
nomain was used and no alternative glossary has been defined.
            \GlossariesWarning{\string\defglsentryfmt\space attempting to set
                a display style for \string\glsdefaulttype\space but there
                doesn't seem to be a default glossary. If you have used
                'nomain', you need to define an alternative glossary}%
        \else
            \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
        \fi
    }

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

\@gls@forbidtexext As a security precaution, don't allow the user to specify a 'tex' extension for
any of the glossary files. (Just in case a seriously confused novice user doesn't
know what they're doing.) The argument must be a control sequence whose
replacement text is the requested extension.
    \newcommand*\@gls@forbidtexext[1]{%
        \ifboolexpr{test {\ifdefstring{#1}{tex}}
            or test {\ifdefstring{#1}{TEX}}}
        {%
            \def#1{nottex}%
            \PackageError{glossaries}%
                {Forbidden '.tex' extension replaced with '.nottex'}%
                {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
                    Don't use '.tex' as an extension for a temporary file.}%
        }%
        {%
        }%
    }

```

`\gls@gobbleopt` Discard optional argument.

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

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

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

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

`\newglossary`

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

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

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

`\ns@newglossary` Define the unstarred version.

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

Check if default has been set

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

Add this to the list of glossary types:

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

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

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

Store the file extensions:

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



```

\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname

```

Store the title:

```

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

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

```

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

```

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

```

Define sort counter if required:

```

\@gls@defsortcount{#2}%

```

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

```

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

```

`\altnewglossary`

```

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

```

Only define new glossaries in the preamble:

```

\@onlypreamble{\newglossary}

```

Only define new glossaries before `\makeglossaries`

```

\@onlypremakeg\newglossary

```

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

`\@newglossary`

```

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

```

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

```
\@gls@setcounter
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @gls@#1@counter\endcsname{#2}%
  Add counter to xindy list, if not already added:
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
```

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

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

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

```
\glsdefmain
```

Define the “acronym” glossaries if required.

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

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

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

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

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

```

    {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
    }%
    {%
    \protected@eappto\@gls@nohyperlist{, #1}%
    }%
}

```

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

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

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

```

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

```

2.7 Defining new entries

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

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

```

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

```

description The `description` key is usually only used in the glossary, but can be made to appear in the text by redefining `\glsentryfmt` or using `\defglsentryfmt`. The `description` key is required when defining a new glossary entry. If a long description is required, use `\longnewglossaryentry` instead of `\newglossaryentry`.

```

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

```

descriptionplural

```

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

```

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

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

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

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

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

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

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

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

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

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

`\@gls@default@value`

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

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

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

symbolplural

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

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

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

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

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

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

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

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

`\gls@set@xr@key`

Assign a cross-reference key.

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

`\gls@xr@key`

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

`\gls@checkseeallowed`

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

```

\@gls@see@noindex
}

checkseeallowed@preambleonly

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

parent The parent key specifies the parent entry, if required.
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}

nonumberlist The nonumberlist key suppresses or activates the number list for the given entry.

\define@choicekey{glossentry}{nonumberlist}%
[\@gls@nonumberlist@val\@gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\@gls@nonumberlist@nr\relax
    \def\@glo@prefix{\@gls@nonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\@gls@nextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}

\@gls@noidx@nonumberlist \makenoidxglossaries sets \@gls@savenonumberlist to this.
\newcommand*{\@gls@noidx@nonumberlist}[1]{%
  \cslet\@glo@\@glo@label @prenumberlist\@glo@prefix
}%

\@gls@noidx@prenumberlist

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

\@gls@savenonumberlist The nonumberlist option isn't saved by default (as it just sets the prefix) which
isn't a problem when the entries are defined in the preamble, but causes a
problem when entries are defined in the document. In this case, the value needs
to be saved so that it can be written to the .glsdefs file.

\newcommand*{\@gls@savenonumberlist}[1]{%

\@gls@initnonumberlist

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

```

```

\@gls@initnonumberlist
    \newcommand*{\@gls@storenonumberlist}[1]{%

\@gls@enablesavenonumberlist Allow the nonumberlist value to be saved.
    \newcommand*{\@gls@enablesavenonumberlist}{%
    \renewcommand*{\@gls@initnonumberlist}{%
        \undef\@glo@nonumberlist
    }%
    \renewcommand*{\@gls@savenonumberlist}[1]{%
        \def\@glo@nonumberlist{##1}%
    }%
    \renewcommand*{\@gls@storenonumberlist}[1]{%
        \ifdef\@glo@nonumberlist
        {%
            \cslet{glo@glstetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
        }%
        {}%
    }%
    \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}

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

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

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

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

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

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

```

user6

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\@gls@noexpand@field`

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

`\@gls@noexpand@fields`

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

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

```

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

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

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

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

```

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

`\gls@assign@field`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

\AtBeginDocument{\gls@begindocdefs}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

Write key value information:

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

Provide hook:

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

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

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

```

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

```

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

`\@gls@fetchfield`

Fetches the internal field label from the given user *<field>* and stores in *<cs>*.

```
\newcommand*{\@gls@fetchfield}[2]{%
```

Ensure user field name is fully expanded

```
\edef\@gls@thisval{#2}%
```

Iterate through known mappings until we find the one for this field.

```

\@for\@gls@map:=\@gls@keymap\do{%
\edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
\ifdefequal{\@this@key}{\@gls@thisval}%
{%

```

Found it.

```
\edef#1{\expandafter\@secondoftwo\@gls@map}%
```

Break out of loop.

```

\@endfortrue
}%
}%
}%
}

```

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

`\glsaddstoragekey`

Similar to `\glsaddkey` but intended for keys whose values aren't explicitly used in the document, but might be required behind the scenes by other commands.

```
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
```

Starred version switches on expansion for this key.

```

\newcommand*{\@sglsaddstoragekey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%

```

```

\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
  \@@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@gls@saddstoragekey{#1}%
}

```

Unstarred version doesn't override default expansion.

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

Check the specified key doesn't already exist.

```

\key@ifundefined{glossentry}{#1}%
{%

```

Set up the key.

```

\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%

```

Set the default value.

```
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
```

Assignment code.

```

\appto\@newglossaryentryposthook{%
  \letcs{\@glo@tmp}{@glo@#1}%
  \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%

```

Define the no-link commands.

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
}%
\PackageError{glossaries}{Key '#1' already exists}{}%
}%
}

```

```

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

```

\gls@saddkey

Allow user to add their own custom keys.

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

Starred version switches on expansion for this key.

```

\newcommand*{\@sgls@saddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}

```



```

    }%
  }%
  {}%
  \@glsaddkey{#1}%
}

```

Unstarred version doesn't override default expansion.

```
\newcommand*{\@glsaddkey}[7]{%
```

Check the specified key doesn't already exist.

```

  \key@ifundefined{glossentry}{#1}%
  {%

```

Set up the key.

```

  \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
  \appto\@gls@keymap{,{#1}{#1}}%

```

Set the default value.

```
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
```

Assignment code.

```

\appto\@newglossaryentryposthook{%
  \letcs{@glo@tmp}{@glo@#1}%
  \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%

```

Define the no-link commands.

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%

```

Now for the commands with links. First the version with no case change:

```

\ifcsdef{@gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  {}%
}%
{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1@\endcsname}[2][ ]{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[ ]}}%
  \csdef{@gls@user@#1@}##1##2[##3]{%
    \@gls@field@link{##1}{##2}{#3{##2}##3}%
  }%
  \newrobustcmd*{#5}{%
    \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
}%

```

Next the version with the first letter converted to upper case:

```
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
  {}%
}%
{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
  \csdef{@Gls@user@#1@}##1##2[##3]{%
    \@gls@field@link{##1}{##2}{#4{##2}##3}%
  }%
  \newrobustcmd*{#6}{%
    \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
  }%
```

Finally the all caps version:

```
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
  \csdef{@GLS@user@#1@}##1##2[##3]{%
    \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
  }%
  \newrobustcmd*{#7}{%
    \expandafter\@gls@hyp@opt\csname @GLS@user@#1@\endcsname}%
  }%
```

Add mappings.

```
\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{%
  \PackageError{glossaries}{Key '#1' already exists}{}%
}
```

```
}%
}
```

`\glsfieldxdef`

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

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

`\glsfielddedef`

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

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

`\glsfieldgdef`

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

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

```

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

```

\glsfielddef

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

```

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

```

\glsfieldfetch

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

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

```

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

```

\glsunexpandedfieldvalue

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

For use in expandable contexts where the field value is required, but the contents should not be expanded. The $\langle field \rangle$ argument must be the internal field name.

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

```
\ifglsfieldeq{<label>}{<field>}{<string>}{<true>}{<false>}
```

\backslash ifglsfieldeq

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

```
\newrobustcmd{\ifglsfieldeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
```

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

\backslash ifglsfielddefeq

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

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

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

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

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

\ifglscfieldcseq

As above but uses \ifcsstrequal instead of \ifdefstrequal

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

\glswritedefhook

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

\glscassign@desc

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

\longnewglossaryentry

```
\newcommand{\longnewglossaryentry}[3]{%
  \glscdoifnoexists{#1}%
  {%
    \bgroup
    \let\org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \org@newglossaryentryprehook
    }%
    \renewcommand*{\glscassign@desc}[1]{%
      \global\cslet{glo@\glscdetoklabel{#1}@desc}{\glo@desc}%
      \global\cslet{glo@\glscdetoklabel{#1}@descplural}{\glo@desc}%
    }
    \glscdefglossaryentry{#1}{#2}%
  }
}
```

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

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

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

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

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

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

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

Prevent any further use of `\GlsSetQuote`:

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

Store label

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Set the default sort:

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

Set the default counter:

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

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

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

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

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

```
\@gls@initnonnumberlist
```

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

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

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

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

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

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

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

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

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

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

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

```
\@newglossaryentryprehook
```

Extract key-val information from third parameter:

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

Check there is a default glossary.

```
\ifundef\glsdefaulttype
```

```
{%
```

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

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

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

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

```
}%
```

```
{}%
```

Assign type. This must be fully expandable

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

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

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

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

```
{%
```

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

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

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

```
in it}%
```

```
}%
```

```
{%
```

Check if it's an ignored glossary

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

```
{%
```


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

```

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

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

```

Initialise level to 0.

```
\gls@level=0\relax
```

Has this entry been assigned a parent?

```
\ifx\@glo@parent\@empty
```

Doesn't have a parent. Set `\glo@<label>@parent` to empty.

```

\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else

```

Has a parent. Check to ensure this entry isn't its own parent.

```

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

```

Check the parent exists:

```

\ifglentryexists{\@glo@parent}%
{%

```

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

```

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

```

Determine level.

```

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

```

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

```

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

```

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

```

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

```

```

\fi
\fi
}%
{%

```

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

```

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

```

Set the level for this entry

```

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

```

Define commands associated with this entry:

```

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

```

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

```

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

\ifcsundef{glotype@\@glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%

```

```

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

```

Set default numberlist if not defined:

```

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

```

Store nonumberlist setting if we're in the document environment.

```

\@gls@storenonumberlist{\@glo@label}%

```

The smaller and smallcaps options set the description to \@glo@first. Need to check for this, otherwise it won't get expanded if the description gets sanitized.

```

\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \@glsnodesc
  \let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%

```

Set the sort key for this entry:

```

\@gls@defsort{\@glo@type}{\@glo@label}%

\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi

```

```

\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
{\csname glo@\@glo@label @symbol\endcsname}
{\@glo@label}{symbolplural}{\@glo@symbolplural}%

```

Define an associated boolean variable to determine whether this entry has been used yet (needs to be defined globally):

```

\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname

```

Sort out any cross-referencing if required.

```

\@glo@autosee

```

Determine and store main part of the entry's index format.

```

\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%

```

Define entry counters if enabled:

```

\@newglossaryentry@defcounters

```

Add end hook in case another package wants to add extra keys.

```

\@newglossaryentryposthook
}

```

`\@glo@autosee` Automatically implement `\glssee`.

```

\newcommand*{\@glo@autosee}{%
\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{,}%
  \forglseentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\@glo@thislabel\glo@label
      {\protected@eappto\glo@list{\glo@label,}}%
    }%
  \cslet\glolist@\glo@type{\glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}

\@glossaryentryfield Indicate what command should be used to display each entry in the glossary.
(This enables the glossaries-accsupp package to use \accsuppglossaryentryfield
instead.)
\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@ifnotmeasuring`

```
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}%
  {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
  }%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
  \ifmeasuring@
  \else
    #1%
  \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
```

`\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{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
  }
```

```

\let\glspatchtabularx\relax
}%
{}%
}

```

`\glsreset` The command `\glsreset{<label>}` can be used to set the entry flag to indicate that it hasn't been used yet. The required argument is the entry label.

```

\newcommand*\glsreset[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}

```

`\glslocalreset` As above, but with only a local effect:

```

\newcommand*\glslocalreset[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}

```

`\glsunset` The command `\glsunset{<label>}` can be used to set the entry flag to indicate that it has been used. The required argument is the entry label.

```

\newcommand*\glsunset[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsunset{#1}%
}%
}%
}

```

`\glslocalunset` As above, but with only a local effect:

```

\newcommand*\glslocalunset[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalunset{#1}%
}%
}%
}

```



```

\@glslocalunset Local unset. This defaults to just \@glslocalunset but is changed by
\glsenableentrycount.
\newcommand*{\@glslocalunset}{\@glslocalunset}

\@@glslocalunset Local unset without checks.
\newcommand*{\@@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}

\@glsunset Global unset. This defaults to just \@glsunset but is changed by \glsenableentrycount.

\newcommand*{\@glsunset}{\@glsunset}

\@@glsunset Global unset without checks.
\newcommand*{\@@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}

\@glslocalreset Local reset. This defaults to just \@glslocalreset but is changed by
\glsenableentrycount.
\newcommand*{\@glslocalreset}{\@glslocalreset}

\@@glslocalreset Local reset without checks.
\newcommand*{\@@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}

\@glsreset Global reset. This defaults to just \@glsreset but is changed by \glsenableentrycount.

\newcommand*{\@glsreset}{\@glsreset}

\@@glsreset Global reset without checks.
\newcommand*{\@@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}

Reset all entries for the named glossaries (supplied in a comma-separated
list). Syntax: \glsresetall[glossary-list]

\glsresetall
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}

```

As above, but with only a local effect:

`\glslocalresetall`

```
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
```

Unset all entries for the named glossaries (supplied in a comma-separated list).

Syntax: `\glsunsetall` [*<glossary-list>*]

`\glsunsetall`

```
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
```

As above, but with only a local effect:

`\glslocalunsetall`

```
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
```

2.9 Keeping Track of How Many Times an Entry Has Been Unset

Version 4.14 introduced `\glsenableentrycount` that keeps track of how many times an entry is marked as used. The counter is reset back to zero when the first use flag is reset. Note that although the word “counter” is used here, it’s not an actual L^AT_EX counter or even an explicit T_EX count register but is just a macro. Any of the commands that use `\glsunset` or `\glslocalunset`, such as `\gls`, will automatically increment this value. Commands that don’t modify the first use flag (such as `\glstext` or `\glsentrytext`) don’t modify this value.

`\newglossaryentry@defcounters` Define entry fields to keep track of how many times that entry has been marked as used.

```
\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
```

`\ifglsresetcurrcount` A conditional to determine whether or not to reset the entry count to zero if the first use flag is reset. The default is now not to do this, which changes the implementation for v4.49 and below.

```
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
```

`\glsenableentrycount` Enables tracking of how many times an entry has been marked as used.

```
\newcommand*\glsenableentrycount}{%
```

Enable new entry fields.

```
\let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
```

Disable `\newglossaryentry` in the document environment.

```
\renewcommand*\gls@defdocnewglossaryentry}{%
\renewcommand*\newglossaryentry[2]{%
\PackageError{glossaries}{\string\newglossaryentry\space
may only be used in the preamble when entry counting has
been activated}{If you use \string\glsenableentrycount\space
you must place all entry definitions in the preamble not in
the document environment}}%
}%
}%
```

Define commands `\glsentrycurrcount` and `\glsentryprevcount` to access these new fields. Default to zero if undefined.

```
\newcommand*\glsentrycurrcount}[1]{%
\ifcsundef{glo@\glsdetoklabel{##1}@currcount}}%
{0}{\@gls@entry@field{##1}{currcount}}}%
}%
\newcommand*\glsentryprevcount}[1]{%
\ifcsundef{glo@\glsdetoklabel{##1}@prevcount}}%
{0}{\@gls@entry@field{##1}{prevcount}}}%
}%
```

Make the unset and reset functions also increment or reset the entry counter.

```
\renewcommand*\@glsunset}[1]{%
\@glsunset{##1}%
\@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
\@glslocalunset{##1}%
\@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
\@glsreset{##1}%
\ifglsresetcurrcount\csgdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
\@glslocalreset{##1}%
\ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
```

Alter behaviour of `\cgl`s. (Only global unset is used if previous count was one as it doesn't make sense to have a local unset here given that the previous count was global.)

```
\def\@cgl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
```

Similarly for the analogous commands. No case change plural:

```
\def\@cglspl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglsplformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@glspl@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase singular:

```
\def\@cGls@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase plural:

```
\def\@cGlspl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@Glspl@{##1}{##2}[##3]%
  \fi
}%
```

Write information to aux file at the end of the document

```
\AtEndDocument{\@gls@write@entrycounts}%
```

Fetch previous count information from aux file. (No check here to determine if the entry is still defined.)

```
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{gls@glsetoklabel{##1}@prevcount}{##2}%
}%
```

`\glsenableentrycount` may only be used once and only in the preamble.

```
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
```

`\@gls@increment@currcount`

```
\newcommand*{\@gls@increment@currcount}[1]{%
\csxdef{glo@\glsdetoklabel{#1}@currcount}{%
\number\numexpr\glsentrycurrcount{#1}+1}%
}
```

`\@gls@local@increment@currcount`

```
\newcommand*{\@gls@local@increment@currcount}[1]{%
\csedef{glo@\glsdetoklabel{#1}@currcount}{%
\number\numexpr\glsentrycurrcount{#1}+1}%
}
```

`\@gls@write@entrycounts` Write the entry counts to the aux file. Use `\immediate` since this occurs right at the end of the document. Only write information for entries that have been used. (Some users have a file containing vast numbers of entries, many of which may not be used. There's no point writing information about the entries that haven't been used and it will only slow things down.)

```
\newcommand*{\@gls@write@entrycounts}{%
\immediate\write\@auxout
{\string\providecommand*\string\@gls@entry@count}[2]{}}%
\forallglsentries{\@glsentry}{%
\ifglsused{\@glsentry}%
{\immediate\write\@auxout
{\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
}%
}%
}
```

`\@gls@entry@count` Default behaviour is to ignore arguments. Activated by `\glsenableentrycount`.

```
\newcommand*{\@gls@entry@count}[2]{}%
```

`\cgl` Define command that works like `\gls` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\gls` but issues a warning.)

```
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
```

`\@cgl` Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@cgl}[2][{}]{%
\new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
```

`\@cgl@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%
}

```

`\cGls` Define command that works like `\Gls` but behaves differently if the entry count function is enabled. (If not enabled, it behaves the same as `\Gls` but issues a warning.)

```

\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}

```

Add sentence case mapping.

```

\glsmfuaddmap{\cgl@s}{\cGls}

```

`\@cGls` Defined the un-starred form. Need to determine if there is a final optional argument

```

\newcommand*{\@cGls}[2][ ]{%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}[]}%
}

```

`\@cGls@` Read in the final optional argument. This defaults to same behaviour as `\Gls` but issues a warning.

```

\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]%
}

```

`\cGlsformat` Format used by `\cGls` if entry only used once on previous run. The first argument is the label, the second argument is the insert text.

```

\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Gls@sentrylong{#1}}{\Gls@sentryfirst{#1}}#2%
}

```

`\cgl@spl` 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*{\cgl@spl}{\@gls@hyp@opt\@cgl@spl}

```

`\@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}{\glsenentrylongpl{#1}}{\glsenentryfirstplural{#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}{\@glshyp@opt\@cGlsp1}

```

Add sentence case mapping.

```

\glsmfuaddmap{\cglsp1}{\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}{\Glsenentrylongpl{#1}}{\Glsenentryfirstplural{#1}}#2%
}

```

2.10 Loading files containing glossary entries

Glossary entries can be defined in an external file. These external files can contain `\newglossaryentry` and `\newacronym` commands.¹

```
\loadglsentries[⟨type⟩]{⟨filename⟩}
```

This command will input the file using `\input`. The optional argument specifies to which glossary the entries should be assigned if they haven't used the `type` key. If the optional argument is not specified, the default glossary is used. Only those entries used in the document (via `\glslink`, `\gls`, `\glspl` and uppercase variants or `\glsadd` and `\glsaddall` will appear in the glossary). The mandatory argument is the filename (with or without `.tex` extension).

```
\loadglsentries
\newcommand*{\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
```

`\loadglsentries` can only be used in the preamble:

```
\@onlypreamble{\loadglsentries}
```

2.11 Using glossary entries in the text

Any term that has been defined using `\newglossaryentry` (or `\newacronym`) can be displayed in the text (i.e. outside of the glossary) using one of the commands defined in this section. Unless you use `\glslink`, the way the term appears in the text is determined by `\glsdisplayfirst` (if it is the first time the term has been used) or `\glsdisplay` (for subsequent use). Any formatting commands (such as `\textbf`) is governed by `\glstextformat`. By default this just displays the *link text* “as is”.

```
\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}{%
```

¹and any other valid L^AT_EX code that can be used in the preamble.


```

\ifdefempty\glscustomtext
{%
  \glusifplural
  {%
Plural form
    \glscapscase
    {%
Don't adjust case
    \ifglused\glslabel
    {%
Subsequent use
    \glentryplural{\glslabel}\glsinsert
    }%
    {%
First use
    \glentryfirstplural{\glslabel}\glsinsert
    }%
    }%
    {%
Make first letter upper case
    \ifglused\glslabel
    {%
Subsequent use.
    \Glsentryplural{\glslabel}\glsinsert
    }%
    {%
First use
    \Glsentryfirstplural{\glslabel}\glsinsert
    }%
    }%
    {%
Make all upper case
    \ifglused\glslabel
    {%
Subsequent use
    \glsupercase{\glentryplural{\glslabel}\glsinsert}%
    }%
    {%
First use
    \glsupercase{\glentryfirstplural{\glslabel}\glsinsert}%
    }%
    }%
    }%
    {%

```

Singular form

```
\glscapscase  
{%
```

Don't adjust case

```
\ifglused\glslabel  
{%
```

Subsequent use

```
\glentrytext{\glslabel}\glinsert  
}%  
{%
```

First use

```
\glentryfirst{\glslabel}\glinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglused\glslabel  
{%
```

Subsequent use

```
\Glentrytext{\glslabel}\glinsert  
}%  
{%
```

First use

```
\Glentryfirst{\glslabel}\glinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglused\glslabel  
{%
```

Subsequent use

```
\gluppercase{\glentrytext{\glslabel}\glinsert}%  
}%  
{%
```

First use

```
\gluppercase{\glentryfirst{\glslabel}\glinsert}%  
}%  
}%  
}%  
}%  
{%
```

Custom text provided in `\gldisp`. (The insert is most likely to be empty at this point.)

```

        \glscustomtext\glsinsert
    }%
}

```

`\glsgenacfmt` Define a generic acronym format that uses the long and short keys (or their plurals) and `\acrfullformat`, `\firstacronymfont` and `\acronymfont`.

```

\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%

```

Subsequent use:

```

        \glsifplural
    {%

```

Subsequent plural form:

```

        \glscapscase
    {%

```

Subsequent plural form, don't adjust case:

```

        \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
    }%
    {%

```

Subsequent plural form, make first letter upper case:

```

        \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
    }%
    {%

```

Subsequent plural form, all caps:

```

        \glssupercase
        {\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:

```
\glsupercase
  {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
}%
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glscapscase
{%
```

First use plural form, don't adjust case:

```
\genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, make first letter upper case:

```
\Genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, all caps:

```
\glsupercase
  {\genplacrformat{\glslabel}{\glsinsert}}}%
}%
}%
{%
```

First use singular form

```
\glscapscase
{%
```

First use singular form, don't adjust case:

```
\genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, make first letter upper case:

```
\Genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, all caps:

```
\glsupercase
  {\genacrformat{\glslabel}{\glsinsert}}}%
}%
}%
}%
```

```

    }%
    {%
User supplied text.
    \glscustomtext
    }%
}

```

`\genacrfullformat`

```
\genacrfullformat{<label>}{<insert>}
```

The full format used by `\glsgenacfmt` (singular).

```

\newcommand*{\genacrfullformat}[2]{%
  \glentrylong{#1}#2\space
  (\protect\firstacronymfont{\glentryshort{#1}})%
}

```

`\Genacrfullformat`

```
\Genacrfullformat{<label>}{<insert>}
```

As above but makes the first letter upper case.

```

\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}

```

`\genplacrfullformat`

```
\genplacrfullformat{<label>}{<insert>}
```

The full format used by `\glsgenacfmt` (plural).

```

\newcommand*{\genplacrfullformat}[2]{%
  \glentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glentryshortpl{#1}})%
}

```

`\Genplacrfullformat`

```
\Genplacrfullformat{<label>}{<insert>}
```

As above but makes the first letter upper case.

```

\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}

```

`\glsdisplayfirst` `\glsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50.
Use rollback if required.

`\glsdisplay` `\glsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplay` `\defglsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplayfirst` `\defglsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

2.11.1 Links to glossary entries

The links to glossary entries all have a first optional argument that can be used to change the format and counter of the associated entry number. Except for `\glslink` and `\glsdisp`, the commands like `\gls` have a final optional argument that can be used to insert additional text in the link (this will usually be appended, but can be redefined using `\defglsentryfmt`). It goes against the L^AT_EX norm to have an optional argument after the mandatory arguments, but it makes more sense to write, say, `\gls{label}[’s]` rather than, say, `\gls[append=’s]{label}`. Since these control sequences are defined to include the final square bracket, spaces will be ignored after them. This is likely to lead to confusion as most users would not expect, say, `\gls{<label>}` to ignore following spaces, so `\new@ifnextchar` from the `amsgen` package is required.

The following keys can be used in the first optional argument. The `counter` key checks that the value is the name of a valid counter.

```
\define@key{glslink}{counter}{%
  \ifcsundef{c#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
```

The value of the `format` key should be the name of a command (without the initial backslash) that has a single mandatory argument which can be used to format the associated entry number.

```
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
```

```
s@glslink@hyper@update@hook
```

```
\newcommand*{\@gls@glslink@hyper@update@hook}{}
```

The `hyper` key is a boolean key, it can either have the value `true` or `false`, and indicates whether or not to make a hyperlink to the relevant glossary entry. If `hyper` is `false`, an entry will still be made in the glossary, but the given text won't be a hyperlink.

```
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
```

Initialise `hyper` key.

```
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
```

Commands like `\gls` and `\glsdisp` unset the first use flag afterwards. Provide a way to localise this.

```
\@gls@save@glslocal
```

```
\newcommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
```

```
\@gls@restore@glslocal
```

```
\newcommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset\glslocalunset
  \else
    \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
```

```
\@gls@do@glsunset
```

```
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
```

The `local` key is a boolean key. If `true` this indicates that commands such as `\gls` should only do a local reset rather than a global one.

```
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
```

The original `\glsifhyper` command isn't particularly useful as it makes more sense to check the actual hyperlink setting rather than testing whether the starred or unstarred version has been used. Therefore, as from version 4.08, `\glsifhyper` is deprecated in favour of `\glsifhyperon`. In case there is a particular need to know whether the starred or unstarred version was used, provide a new command that determines whether the `*`-version, `+`-version or unmodified version was used.

```
\glslinkvar{<unmodified case>}{<star case>}{<plus case>}
```

`\glslinkvar` Initialise to unmodified case.

```
\newcommand*{\glslinkvar}[3]{#1}
```

`\glsifhyper` `\glsifhyper` deprecated in v4.08 (2014-07-30) and removed in v4.50.

`\@gls@hyp@opt` Used by the commands such as `\glslink` to determine whether to modify the hyper option.

```
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
```

`\s@gls@hyp@opt` Starred version

```
\newcommand*{\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
```

`\p@gls@hyp@opt` Plus version

```
\newcommand*{\p@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
```

Syntax:

```
\glslink[<options>]{<label>}{<text>}
```

Display *<text>* in the document, and add the entry information for *<label>* into the relevant glossary. The optional argument should be a key value list using the `\glslink` keys defined above.

There is also a starred version:

```
\glslink*[<options>]{<label>}{<text>}
```

which is equivalent to `\glslink[hyper=false,<options>]{<label>}{<text>}`

First determine which version is being used:

`\glslink`

```
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@link
}
```

`\@gls@link` The main part of the business is in `\@gls@link` which shouldn't check if the term is defined as it's called by `\gls` etc which also perform that check.

```
\newcommand*{\@gls@link}[3][]{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
```


Display the specified text. (The entry doesn't exist so there's nothing to link it to.)

```
\glstextformat{#3}%
}%

\glspostlinkhook
}
```

`\Glslink` Provide a sentence-case version in order to set up a map.

```
\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@@link
}

\newcommand*{\@Gls@@link}[3][{}]{%
\@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
```

`\glspostlinkhook`

```
\newcommand*{\glspostlinkhook}{}
```

`\@gls@link@checkfirsthyper` Check for first use and switch off hyper key if hyperlink not wanted. (Should be off if first use and hyper=false is on or if first use and both the entry is in an acronym list and the acrf footnote 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
\ifglsacrf footnote
\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@gl:disablehyperinlist Disable hyperlink if in the “nohyper” list.
    \newcommand*{\do@gl:disablehyperinlist}{%
        \expandafter\DTLifinlist\expandafter{\gls@type}{\@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\gls@type{\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@gl: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@gl@link@checkfirsthyper
        \setkeys{glslink}{#1}%

```

Add a hook for the user to customise things after the keys have been set.

```
\glslinkpostsetkeys
```

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

```
\@gls@setsort{\glslabel}%
```

(De-tok'ing done by `\@do@wrglossary`)

```
\glsencapwrcontent{\@do@wrglossary{#2}}%
```

```
\ifKV@glslink@hyper
```

```
\@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
```

```
\else
```

```
\glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
```

```
\fi
```

Restore original setting

```
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
```

```
\@gls@restore@glslocal
```

```
}
```

`\glsencapwrcontent` This command is also provided by glossaries-extra v1.49. It's purpose is to resolve [issue #189](#).

```
\newcommand*{\glsencapwrcontent}[1]{\@#1}
```

```
\glolinkprefix
```

```
\newcommand*{\glolinkprefix}{glo:}
```

`\glsentrycounter` Set default value of entry counter

```
\def\glsentrycounter{\glscounter}%
```

`\@gls@saveentrycounter` Need to check if using equation counter in align environment:

```
\newcommand*{\@gls@saveentrycounter}{%
```

```
\def\@gls@Hcounter}{%
```

Are we using equation counter?

```
\ifthenelse{\equal{\@gls@counter}{equation}}{%
```

```
{
```

If we're in align environment, `\xatlevel@` will be defined. (Can't test for `\@currenvir` as may be inside an inner environment.)

```
\ifcsundef{xatlevel@}%
```

```
{%
```

```
\edef\theglsentrycounter{\expandafter\noexpand
```

```
\csname the\@gls@counter\endcsname}%
```

```
}%
```

```
{%
```

```
\ifx\xatlevel@\@empty
```

```
\edef\theglsentrycounter{\expandafter\noexpand
```

```

\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglentrycounter{\csname the\@gls@counter\endcsname}%
Check if hyperref version of this counter
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglentrycounter}%
}%
{%
\def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
}%
\protected@edef\theHglentrycounter{\@gls@Hcounter}%
\restorecounters@
\fi
}%
}%
{%

```

Not using equation counter so no special measures:

```

\edef\theglentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
Check if hyperref version of this counter
\ifx\@gls@Hcounter\@empty
\ifcsundef{theH\@gls@counter}%
{%
\def\theHglentrycounter{\theglentrycounter}%
}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
}

```

`\@set@glo@numformat` Set the formatting information in the format required by `makeindex`. The first argument is the format specified by the user (via the format key), the second argument is the name of the counter used to indicate the location, the third argument is a control sequence which stores the required format and the fourth argument (new to v3.0) is the hyper-prefix.

```

\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mکیدxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%

```

```

    \@gls@checkmkidxchars#1%
}

```

Check to see if the given string starts with a (or). If it does set \@glo@prefix to the starting character, and \@glo@suffix to the rest (or glsnumberformat if there is nothing else), otherwise set \@glo@prefix to nothing and \@glo@suffix to all of it.

```

\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1(\relax
  \def\@glo@prefix{(%
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \if#1)\relax
    \def\@glo@prefix{)%
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}

```

\@gls@escbsdq Escape backslashes and double quote marks. The argument must be a control sequence.

```

\newcommand*{\@gls@escbsdq}[1]{%
  \def\@gls@checkedmkidx{}%
  \let\gls@xdystring=#1\relax
  \@onelevel@sanitize\gls@xdystring
  \edef\do@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\null}%
  \do@gls@xdycheckbackslash
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \def\@gls@checkedmkidx{}%
  \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
}

```

Unsanitize \@gls@numberpage, \@gls@alphpage, \@gls@Alphpage and \@gls@romanpage (thanks to David Carlisle for the suggestion.) Skip if esclocations=false.

```

\ifglsecllocations
  \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@xdyststring
  {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\fi

```

Assign to required control sequence

```

\let#1=\gls@xdyststring
}

```

Catch special characters (argument must be a control sequence):

\@gls@checkmkidxchars

```

\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglxsindy
    \@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
}

```

```

\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
  \fi
}

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\@null#3\@null
\def\@gls@checkescquote{\@gls@checkescquote#2\@"\@null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\#"#3\@null}%
\fi
\fi
\fi
\@gls@checkescquote
}

```

\@gls@checkescactual Similarly for \? (which is replaces @ as makeindex's special character):

```

\def\@gls@checkescactual#1\?#2\?#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\@null#3\@null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\@null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\@null}%
\fi
\fi
\@gls@checkescactual
}

```

\@gls@checkescbar Similarly for \|:

```

\def\@gls@checkescbar#1\|#2\|#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%

```



```

\def\@gls@checkescbar{\@gls@checkescbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
\else
\def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
\fi
\fi
\@gls@checkescbar
}

```

\@gls@checkesclevel Similarly for \!:

```

\def\@gls@checkesclevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesclevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}

```

\@gls@checkbar and for |:

```

\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

\gls@quotechar\gls@encapchar\gls@quotechar\gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}

```

\gls@checklevel and for !:

```

\def\@gls@checklevel#1!#2!#3\null{%
\gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@levelchar\gls@quotechar\gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}

```

\gls@checkactual and for ?:

```

\def\@gls@checkactual#1?#2?#3\null{%
\gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}

```

\@gls@checkesc and for \:

```

\bgrou
\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
}
\egrou

```

\@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}%
  {%
    \settoheight{\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
}

```

\@glstarget If \hypertarget is not defined, \@glstarget ignores its first argument and just does the second argument, otherwise it is equivalent to \hypertarget.

```

\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}

```

Glossary hyperlinks can be disabled using `\glsdisablehyper` (effect can be localised):

```
\glsdisablehyper
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
```

Glossary hyperlinks can be enabled using `\glsenablehyper` (effect can be localised):

```
\glsenablehyper
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
```

Provide some convenience commands if not already defined:

```
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
```

Syntax:

```
\gls[<options>]{<label>}[<insert text>]
```

Link to glossary entry using singular form. The link text is taken from the value of the `text` or `first keys` used when the entry was defined.

The first optional argument is a key-value list, the same as `\glslink`, the mandatory argument is the entry label. After the mandatory argument, there is another optional argument to insert extra text in the link text (the location of the inserted text is governed by `\glsdisplay` and `\glsdisplayfirst`). As with `\glslink` there is a starred version which is the same as the unstarred version but with the `hyper` key set to `false`. (Additional options can also be specified in the first optional argument.)

First determine which version is being used:

```
\gls
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\@gls
\newcommand*{\@gls}[2][ ]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
```

`\@gls@` Read in the final optional argument:

```
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
  }%
  \def\glsinsert{#3}%
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\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.

```
\glsmfuaddmap{\gls}{\Gls}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Gls}[2][ ]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[ ]}%
}
```

`\@Gls@` Read in the final optional argument:

```
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```

\let\glsifplural\@secondoftwo
\let\glsifscapscase\@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 `\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}%
}%

\gls@postlinkhook
}

```

`\GLS` behaves like `\gls`, but the link text is converted to uppercase:

`\GLS`

```

\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\gls@fublocker{\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@glss@link@checkfirsthyper\@glss@link@checkfirsthyper

\let\glsifplural\@secondoftwo
\let\glsifscapscase\@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@glsunset{#2}%
}%

\glspostlinkhook
}
```

`\glspl` behaves in the same way as `\gls` except it uses the plural form.

`\glspl`

```
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glspl}[2][{}]{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}[]}%
}
```

`\@glspl@` Read in the final optional argument:

```
\def\@glspl@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

\let\glsifplural\@firstoftwo
\let\glsifcaps\@firstofthree
\let\glsifcustom\@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 `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

`\Glspl` behaves in the same way as `\glspl`, except that the first letter of the link text is converted to uppercase (as with `\Gls`, if the first letter has an accent, it will need to be grouped).

`\Glspl`

```
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsp1}{\Glspl1}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glspl1}[2] [] {%
  \new@ifnextchar[{\@Glspl1@{#1}{#2}}{\@Glspl1@{#1}{#2} []}%
}
```

`\@Glspl1@` Read in the final optional argument:

```
\def\@Glspl1@#1#2[#3] {%
  \glsdofexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper

    \let\gl@ifplural\@firstoftwo
    \let\gl@scapscase\@secondofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%

```

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 `\@gl@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gl@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gl@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gl@do@gl@unset{#2}%
}%

\glspostlinkhook
}
```

`\GLSp1` behaves like `\glsp1` except that all the link text is converted to uppercase.

`\GLSp1`

```
\newrobustcmd*{\GLSp1}{\@gl@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@gl@link@checkfirsthyper\@gl@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
  }%
  \def\glsinsert{#3}%
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gl@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gl@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gl@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gl@do@gl@unset{#2}%
}%

\glspostlinkhook
}
```

`\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}{\@gl@hyp@opt\@glsdisp}
```

Defined the un-starred form.

`\@glsdisp`

```
\newcommand*{\@glsdisp}[3][]{%
  \glsdoifexists{#2}{%

    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \def\glscustomtext{#3}%
    \def\glsinsert{}%
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gl@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}{\@gls@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glssunset{#2}%
}%
```

```
\glspostlinkhook
}
```

`\Glsdisp` Provide a sentence-case version to make it easier to map.

```
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
```

`\@Glsdisp`

```
\newcommand*{\@Glsdisp}[3][]{%
\@glsdisp[#1]{#2}{\glssentencecase{#3}}%
}
\glsmfuaddmap{\glsdisp}{\@Glsdisp}
```

`\@gls@link@nocheckfirsthyper` Instead of just setting `\do@gls@link@checkfirsthyper` to `\relax` in `\@gls@field@link`, set it to `\@gls@link@nocheckfirsthyper` in case some other action needs to take place.

```
\newcommand*{\@gls@link@nocheckfirsthyper}{}%
```

`\@gls@field@link`

```
\@gls@field@link{<options>}{<label>}{<link text>}
```

```
\newcommand{\@gls@field@link}[3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\@gls@link[#1]{#2}{#3}%
}%
\glspostlinkhook
}
```

`\glstext` behaves like `\gls` except it always uses the value given by the `text` key and it doesn't mark the entry as used.

`\glstext`

```
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glstext}[2][]{%
\new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstentrytext{#2}#3}%
}
```

`\GLStext` behaves like `\glstext` except the text is converted to uppercase.

`\GLStext`

```
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\glsmfublocker{\GLStext}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLStext}[2][]{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssupercase{\glstentrytext{#2}#3}}%
}
```

`\Glstext` behaves like `\glstext` except that the first letter of the text is converted to uppercase.

`\Glstext`

```
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glstext}{\Glstext}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glstext}[2][]{%
  \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
}
```

`\glsfirst` behaves like `\gls` except it always uses the value given by the first key and it doesn't mark the entry as used.

`\glsfirst`

```
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsfirst}[2][]{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glentryfirst{#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}{\glentryfirst{#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}{\glssupercase{\glentryfirst{#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][\@Glsplural@#1]{%
  \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][\@GLSplural@#1]{%
  \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][\@glsfirstplural@#1]{%
  \new@ifnextchar[\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}}}
```

Read in the final optional argument:

```
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

```
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsfirstplural}[2][{}]{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
```

`\GLSfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\GLSfirstplural`

```
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\glsmfublocker{\GLSfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSfirstplural}[2][{}]{%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryfirstplural{#2}#3}%
}
```

`\glsname` behaves like `\gls` except it always uses the value given by the name key and it doesn't mark the entry as used.

`\glsname`

```
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsname}[2][{}]{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2}[]}}
```


Read in the final optional argument:

```
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
```

`\Glsname` behaves like `\glsname` except that the first letter is converted to uppercase.

`\Glsname`

```
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsname}{\Glsname}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsname}[2][{}]{%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
```

`\GLSname` behaves like `\glsname` except that the link text is converted to uppercase.

`\GLSname`

```
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\glsmfublocker{\GLSname}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSname}[2][{}]{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryname{#2}#3}}%
}
```

`\glsdesc` behaves like `\gls` except it always uses the value given by the description key and it doesn't mark the entry as used.

`\glsdesc`

```
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsdesc}[2][{}]{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
```

`\Glsdesc` behaves like `\glsdesc` except that the first letter is converted to uppercase.

`\Glsdesc`

```
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsdesc}{\Glsdesc}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsdesc}[2] [] {%
  \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
```

`\GLSdesc` behaves like `\glsdesc` except that the link text is converted to uppercase.

`\GLSdesc`

```
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\glsmfublocker{\GLSdesc}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%
}
```

`\glsdescplural` behaves like `\gls` except it always uses the value given by the `descriptionplural` key and it doesn't mark the entry as used.

`\glsdescplural`

```
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\glsdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
```

`\Glsdescplural` behaves like `\glsdescplural` except that the first letter is converted to uppercase.

`\Glsdescplural`

```
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsdescplural}[2][ ]{%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
```

`\GLSdescplural` behaves like `\glsdescplural` except that the link text is converted to uppercase.

`\GLSdescplural`

```
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSdescplural}[2][ ]{%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydescplural{#2}#3}}%
}
```

`\glssymbol` behaves like `\gls` except it always uses the value given by the symbol key and it doesn't mark the entry as used.

`\glssymbol`

```
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glssymbol}[2][ ]{%
  \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2}[ ]}}
```

Read in the final optional argument:

```
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
}
```

`\Glsymbol` behaves like `\glssymbol` except that the first letter is converted to uppercase.

`\Glsymbol`

```
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbol}{\Glsymbol}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsymbol}[2][\@Glsymbol@{#1}{#2}]{\@Glsymbol@{#1}{#2}[]}
```

Read in the final optional argument:

```
\def\@Glsymbol@#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][\@GLSsymbol@{#1}{#2}]{\@GLSsymbol@{#1}{#2}[]}
```

Read in the final optional argument:

```
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssupercase{\glstentrysymbol{#2}#3}%
}
```

`\glssymbolplural` behaves like `\gls` except it always uses the value given by the `symbolplural` key and it doesn't mark the entry as used.

`\glssymbolplural`

```
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glssymbolplural}[2][\@glssymbolplural@{#1}{#2}]{\@glssymbolplural@{#1}{#2}[]}
```

Read in the final optional argument:

```
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstrysymbolplural{#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}{\glstrysymbolplural{#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}{\glssuppercase{\glstrysymbolplural{#2}#3}}%
}
```

`\glsuseri` behaves like `\gls` except it always uses the value given by the `user1` key and it doesn't mark the entry as used.

`\glsuseri`

```
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseri}[2] []{%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
```

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

```
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseri}{\Glsuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuseri}[2][{}]{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
```

`\GLSuseri` behaves like `\glsuseri` except that the link text is converted to uppercase.

`\GLSuseri`

```
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfublocker{\GLSuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseri}[2][{}]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}%
}
```

`\glsuserii` behaves like `\gls` except it always uses the value given by the `user2` key and it doesn't mark the entry as used.

`\glsuserii`

```
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuserii}[2][{}]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
```

\Glsuserii behaves like \glsuserii except that the first letter is converted to uppercase.

\Glsuserii

```
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserii}{\Glsuserii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuserii}[2][]{%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
```

\GLSuserii behaves like \glsuserii except that the link text is converted to uppercase.

\GLSuserii

```
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuserii}[2][]{%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}%
}
```

\glsuseriii behaves like \gls except it always uses the value given by the user3 key and it doesn't mark the entry as used.

\glsuseriii

```
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseriii}[2][]{%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}]
```

Read in the final optional argument:

```
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstryuseriii{#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}{\glstryuseriii{#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}{\glssupercase{\glstryuseriii{#2}#3}}%
}
```

`\glsuseriv` behaves like `\gls` except it always uses the value given by the `user4` key and it doesn't mark the entry as used.

`\glsuseriv`

```
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseriv}[2] []{%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]}
```


Read in the final optional argument:

```
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
```

`\Glsuseriv` behaves like `\glsuseriv` except that the first letter is converted to uppercase.

`\Glsuseriv`

```
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuseriv}[2][{}]{%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
```

`\GLSuseriv` behaves like `\glsuseriv` except that the link text is converted to uppercase.

`\GLSuseriv`

```
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseriv}[2][{}]{%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
```

`\glsuserv` behaves like `\gls` except it always uses the value given by the `user5` key and it doesn't mark the entry as used.

`\glsuserv`

```
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuserv}[2][{}]{%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
```

`\Glsuserv` behaves like `\glsuserv` except that the first letter is converted to uppercase.

`\Glsuserv`

```
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserv}{\Glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuserv}[2][{}]{%
  \new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
```

`\GLSuserv` behaves like `\glsuserv` except that the link text is converted to uppercase.

`\GLSuserv`

```
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuserv}[2][{}]{%
  \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}%
}
```

`\glsuservi` behaves like `\gls` except it always uses the value given by the `user6` key and it doesn't mark the entry as used.

`\glsuservi`

```
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuservi}[2][{}]{%
  \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
```

`\Glsuservi` behaves like `\glsuservi` except that the first letter is converted to uppercase.

`\Glsuservi`

```
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuservi}{\Glsuservi}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsuservi}[2][]{%
  \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}[]}}
```

Read in the final optional argument:

```
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
```

`\GLSuservi` behaves like `\glsuservi` except that the link text is converted to uppercase.

`\GLSuservi`

```
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuservi}[2][]{%
  \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]
```

Read in the final optional argument:

```
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
```

Now deal with acronym related keys. First the short form:

`\acrshort`

```
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@ns@acrshort}[2][]{%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2}[]}]%
}
```

Read in the final optional argument:

```
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
  }
```

Call `\@gl@link` Note that `\@gl@link` sets `\glstype`.

```
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\Acrshort`

```
\newrobustcmd*{\Acrshort}{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
  }
```

Call `\@gl@link` Note that `\@gl@link` sets `\glstype`.

```
\@gl@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\glsupcase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
    }%

```

Call \@gls@link Note that \@gls@link sets \glsstyle.

```

  \@gls@link[#1]{#2}{\csname gls@\glsstyle @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\glsifcaps\@firstoftwo
\let\glsinsert\@empty
\def\glsentryshortpl{#2}#3%
}%

```

Call \@gls@link Note that \@gls@link sets \gls@type.

```

\@gls@link[#1]{#2}{\csname gls@\gls@type @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\glsifcaps\@secondoftwo
\let\glsinsert\@empty
\def\glsentryshortpl{#2}#3%
}%
}

```

Call \@gls@link Note that \@gls@link sets \gls@type.

```

\@gls@link[#1]{#2}{\csname gls@\gls@type @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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
  }%
}

```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```

\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

```

\acrlong

`\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}`

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@acrlong}[2][\%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
```

Read in the final optional argument:

```
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
\def\glscustomtext{%
  \glentrylong{#2}#3%
}%
```

Call \@gls@link Note that \@gls@link sets \glstype.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

\Acrlong

```
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrlong}{\Acrlong}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
```

Read in the final optional argument:

```
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
\def\glscustomtext{%
  \Glsentrylong{#2}#3%
}%
```

Call \@gls@link. Note that \@gls@link sets \glstype.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

\ACRlong

```
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\glsmfublocker{\ACRlong}
```


Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@ACRlong}[2] [] {%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2} []}%
}
```

Read in the final optional argument:

```
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsupcase\@thirdofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont only designed for short form).

```
\def\glscustomtext{%
  \glsupercase{\glsentrylong{#2}{#3}}%
}%
```

Call \@gls@link. Note that \@gls@link sets \glstype.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

Short plural:

\acrlongpl

```
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@acrlongpl}[2] [] {%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2} []}%
}
```

Read in the final optional argument:

```
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsupcase\@firstofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%
  \glentrylongpl{#2}#3%
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\Acrlongpl`

```
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@ns@Acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
```

Read in the final optional argument:

```
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%
  \Glsentrylongpl{#2}#3%
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\ACRlongpl`

```
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@ACRlongpl}[2][ ]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[ ]}%
}
```

Read in the final optional argument:

```
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsupercase\@thirdofthree
    \let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%
  \glsupercase{\glsentrylongpl{#2}{#3}}%
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

2.11.2 Displaying entry details without adding information to the glossary

These commands merely display entry information without adding entries in the associated file or having hyperlinks.

`\@gls@entry@field` Generic version.

```
\@gls@entry@field{\<label>}{\<field>}
```

Now using `\csuse` instead of `\csname` to expand to nothing rather than `\relax` if undefined.

```
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{gls@glsdetoklabel{#1}{#2}}%
}
```

```
\glsletentryfield{\<cs>}{\<label>}{\<field>}
```

`\glsletentryfield`

```

\newcommand*\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}

```

`\@Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{<label>}{<field>}
```

Check for PDF bookmarks performed by `\@Glsentryfield`.

```

\newrobustcmd*\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\glsentencecase\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ??%
  }%
}

```

Aside from `\Glsentryname`, the `\Glsentry...` sentence case commands can all follow the same format.

```
\@Glsentryfield{<entry-label>}{<field-label>}
```

`\@Glsentryfield`

```

\newcommand*\@Glsentryfield}[2]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}

```

`\glstexorpdfstring` Provide a command that will use `\texorpdfstring` if it has been defined. (Remove all the cumbersome tests to determine whether or not that command has been defined.)

```

\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}

```

Get the entry name (as specified by the `name` key when the entry was defined). The argument is the label associated with the entry. Note that unless

you used `name=false` in the `sanitize` package option you may get unexpected results if the `name` key contains any commands.

`\glsentryname`

```
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
```

The sentence-case equivalents were made robust since they use the non-expandable `\makefirstuc`. These will now expand in the PDF bookmarks to sentence case using `\MFUsentencecase` (which is expandable). Obviously, if any of the fields contain non-expandable content (such as acronym font changing commands) then that will cause a problem (which can be dealt with by using `hyperref`'s `\pdfstringdefDisableCommands`).

`\Glsentryname`

```
\newcommand*{\Glsentryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryname}{\Glsentryname}
```

`\@Gls@entryname` This is a workaround in the event that the user defies the warning in the manual about not using `\Glsname` or `\Glsentryname` with acronyms. The check for PDF bookmarks is in `\Glsentryname`. First the default behaviour:

```
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
```

`\@Gls@acrentryname` Now the behaviour when `\setacronymstyle` is used. The original definition of this command used `mfirstuc` internal commands that are now being deprecated.

```
\newrobustcmd*{\@Gls@acrentryname}[1]{%
  \ifglshaslong{#1}%
  {%
```

Just need to ensure that `\acrnameformat` is expanded, which may cause problems as non-robust semantic commands may expand to content that interferes with sentence-casing.

```
    \protected@edef\@glo@text{\@gls@entry@field{#1}{name}}%
    \expandafter\glssentencecase\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` 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{\glsentryfirst}{\Glsentryfirst}
```


Get the plural form (as specified by the `firstplural` key when the entry was defined).

```
\glsentryfirstplural
\newcommand*\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
```

```
\Glsentryfirstplural
\newcommand*\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryfirstplural}{\Glsentryfirstplural}
```

`\glscapitalisewords` Make it easier to switch from `\capitalisewords` to `\capitalisefmtwords`.

```
\newcommand{\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
```

`\glsentrytitlecase` Now uses sentence case in PDF bookmark.

```
\newcommand*\glsentrytitlecase}[2]{%
  \glstexpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
```

`\@glsentrytitlecase`

```
\newrobustcmd*\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
```

Display the glossary type with which this entry is associated (as specified by the `type` key used when the entry was defined). This expands to a label so isn't intended to produce text in the document.

`\glsentrytype`

```
\newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
```

Display the sort text used for this entry. Note that the `sort` key is sanitize, so unexpected results may occur if the `sort` key contained commands. This is likewise not intended to produce text in the document, other than for debugging purposes.

```

\glsentrysort
    \newcommand*\glsentrysort[1]{%
        \@gls@entry@field{#1}{sort}%
    }

\glsentryparent Expands to the label of the entry's parent. This expands to a label so isn't
intended to produce text in the document.
    \newcommand*\glsentryparent[1]{%
        \@gls@entry@field{#1}{parent}%
    }

\glsentryuseri Get the first user key (as specified by the user1 when the entry was defined).
The argument is the label associated with the entry.
    \newcommand*\glsentryuseri[1]{%
        \@gls@entry@field{#1}{useri}%
    }

\Glsentryuseri
    \newcommand*\Glsentryuseri[1]{%
        \@Glsentryfield{#1}{useri}%
    }

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][\glentrytext{\@glo@label}]{%
\def\@glo@label{#2}%
\@glslink{\glo@linkprefix\glsetoklabel{#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@glsetoklabel{#2}@counter\endcsname}%
\setkeys{glossadd}{#1}%

```

Store the entry's counter in `\theglsentrycounter`

```

\@gls@saveentrycounter

```

Define sort key if necessary:

```

\@gls@setsort{#2}%

```

This should use `\@do@wrglossary` rather than `\do@wrglossary` since the whole point of `\glsadd` is to add a line to the glossary.

```

\@do@wrglossary{#2}%
}%
}

```

Identify `\glsadd` as an exclusion, but this will only work if the optional argument is omitted. If the optional argument is required, the `\glsadd` command with arguments will need to be encapsulated with `\MFUskipunc`. See the `mfirstuc v2.08+` manual for further details.

```

\glsmfuexcl{\glsadd}

```

`\@gls@adjustmode`

```
\newcommand*{\@gls@adjustmode}{}  
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
```

The following aren't added as exclusions, since there's no reason for them to occur at the start of content that may require a case-change (such as at the start of a description).

`\glsaddall[<option list>]`

Add all terms defined for the listed glossaries (without displaying any text). If `types` key is omitted, apply to all glossary types.

`\glsaddall`

```
\newrobustcmd*{\glsaddall}[1][]{%  
  \protected@edef\@glo@type{\@glo@types}%  
  \setkeys{glossadd}{#1}%  
  \forallglsentries[\@glo@type]{\@glo@entry}{%  
    \glsadd[#1]{\@glo@entry}%  
  }%  
}
```

`\glsaddallunused[<glossary types>]`

`\glsaddallunused`

Add all used terms defined for the listed glossaries (without displaying any text). If optional argument is omitted, apply to all glossary types. This should typically go at the end of the document.

```
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%  
  \forallglsentries[#1]{\@glo@entry}%  
  {%  
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%  
  }%  
}
```

`\glsignore`

```
\newcommand*{\glsignore}[1]{}%
```

2.13 Creating associated files

The `\writeist` command creates the associated customized `.ist` `makeindex` style file. While defining this command, some characters have their catcodes temporarily changed to ensure they get written to the `.ist` file correctly. The `makeindex` actual character (usually `@`) is redefined to be a `?`, to allow internal commands to be written to the glossary file output file.

The special characters are stored in `\@gls@actualchar`, `\@gls@encapchar`, `\@gls@levelchar` and `\@gls@quotechar` to make them easier to use later, but

don't change these values, because the characters are encoded in the command definitions that are used to escape the special characters (which means that the user no longer needs to worry about `makeindex` special characters).

The symbols and numbers label for group headings are hardwired into the `.ist` file as `glsymbols` and `glsnumbers`, the group titles can be translated (so that `\glsymbolsgroupname` replaces `glsymbols` and `\glsnumbersgroupname` replaces `glsnumbers`) using the command `\glsgetgrouptitle` which is defined in `glossary-hypernav`. This is done to prevent any problem characters in `\glsymbolsgroupname` and `\glsnumbersgroupname` from breaking hyperlinks.

`\glsopenbrace` Define `\glsopenbrace` to make it easier to write an opening brace to a file.
`\edef\glsopenbrace{\expandafter\@gobble\string\{}`

`\glsclosebrace` Define `\glsclosebrace` to make it easier to write an opening brace to a file.
`\edef\glsclosebrace{\expandafter\@gobble\string\}}`

`\glsbackslash` Define `\glsbackslash` to make it easier to write a backslash to a file.
`\edef\glsbackslash{\expandafter\@gobble\string\}`

`\glsquote` Define command that makes it easier to write quote marks to a file in the event that the double quote character has been made active.
`\edef\glsquote#1{\string"#1\string"}`

`\glspcentchar` Define `\glspcentchar` to make it easier to write a percent character to a file.
`\edef\glspcentchar{\expandafter\@gobble\string\%}`

`\glstildechar` Define `\glstildechar` to make it easier to write a tilde character to a file.
`\edef\glstildechar{\string~}`

`\@glsfirstletter` Define the first letter to come after the digits 0,...,9. Only required for `xindy`.
`\ifglsxindy`
`\newcommand*\@glsfirstletter}{A}`
`\fi`

`\GlsSetXdyFirstLetterAfterDigits` Sets the first letter to come after the digits 0,...,9. The starred version sanitizes.
`\newcommand*\GlsSetXdyFirstLetterAfterDigits}{%`
`\@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}`
`\ifglsxindy`
`\newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%`
`\renewcommand*\@glsfirstletter}{#1}}`
`\newcommand*\s@GlsSetXdyFirstLetterAfterDigits[1]{%`
`\renewcommand*\@glsfirstletter}{#1}%`
`\@onelevel@sanitize\@glsfirstletter`
`}`
`\else`
`\newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%`
`\glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}`
`\newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%`

```

        \@GlsSetXdyFirstLetterAfterDigits
    }
\fi

```

`\@xdynumbergrouporder` Specifies the order of the number group.

```

\ifglxsindy
    \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi

```

`\GlsSetXdyNumberGroupOrder` Sets the relative location of the number group. The starred version sanitizes.

```

\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglxsindy
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \renewcommand*{\@xdynumbergrouporder}{#1}%
    }
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
        \renewcommand*{\@xdynumbergrouporder}{#1}%
        \@onelevel@sanitize\@xdynumbergrouporder
    }
\else
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \glsnxindywarning\GlsSetXdyNumberGroupOrder}
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
\fi

```

`\@glsminrange` Define the minimum number of successive location references to merge into a range.

```

\newcommand*{\@glsminrange}{2}

```

`\GlsSetXdyMinRangeLength` Set the minimum range length. The value must either be `none` or a positive integer. The glossaries package doesn't check if the argument is valid, that is left to xindy.

```

\ifglxsindy
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsminrange}{#1}}
\else
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnxindywarning\GlsSetXdyMinRangeLength}
\fi

```

`\writeist`

```

\ifglxsindy

```

Code to use if xindy is required.

```

\def\writeist{%

```

Define write register if not already defined

```
\ifundef{\glswrite}{\newwrite\glswrite}{}%
```

Update attributes list

```
\@gls@addpredefinedattributes
```

Open the file.

```
\openout\glswrite=\listfilename
```

Write header comment at the start of the file

```
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
```

Specify the required styles

```
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{{(require
\string"\@xdystyle.xdy\string")}}%
\fi
}%
```

List the allowed attributes (possible values used by the format key)

```
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
```

Define any additional alphabets

```
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
```

Define location classes.

```
\write\glswrite{^^J; location class definitions^^J}%
```

As from version 3.0, locations are now specified as $\{\langle Hprefix \rangle\}\{\langle number \rangle\}$, so need to add all possible combinations of location types.

```
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
```

Case where $\langle Hprefix \rangle$ is empty:

```
\protected@write\glswrite{{(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{\{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "\}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
```

Nested iteration over all classes:

```
{%
  \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
    \protected\write\glswrite{}\{(define-location-class
      \string"\@gls@classII-\@gls@classI\string"
      ^^J\space\space\space
      (
        :sep "{"
        \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
        :sep "{{"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
    )
  }%
}%
}%
}%
```

User defined location classes (needs checking for new location format).

```
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
```

Cross-reference class. (The unverified option is used as the cross-references are supplied using the list of labels along with the optional argument for `\glsseeformat` which xindy won't recognise.)

```
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
```

Define how cross-references should be displayed. This adds an empty set of braces after the cross-referencing information allowing for the final argument of `\glsseeformat` which gets ignored. (When using `makeindex` this final argument contains the location information which is not required.)

```
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
```

Provide hook to write extra material here (used by `glossaries-extra` to define a `seealso` class).

```
\@xdycrossrefhook
```

List the order to sort the classes.

```
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
```

Specify what to write to the start and end of the glossary file.

```
\write\glswrite{^^J; define the glossary markup^^J}%
```

```

\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%

```

Add all the xindy-only macro definitions (needed to prevent errors in the event that the user changes from xindy to makeindex)

```

\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{}\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%

```

Add the end part of the open tag and the rest of the markup-index information:

```

\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}}%

```

Specify what to put between letter groups

```

\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%

```

Specify what to put between entries

```

\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string"}}%

```

Specify how to format entries

```

\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"}}%

```

Specify how to separate location numbers

```

\write\glswrite{(markup-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{\glpercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glpercentchar\space for document
'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}

\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string

```

```

\glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string\string}\string}\string}
\write\glswrite{delim_n \string\string}\delimN \string}
\write\glswrite{delim_r \string\string}\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
\string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
\string\string}\string\relax
\string\glsgroupresetentrylist \string}
\write\glswrite{symhead_positive \string\glssymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string\gls@suffixFF\string}
\fi

```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the file and disable \writeist.

```

\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi

```

\GlsSetWriteIstHook Allow user to append information to the style file.

```

\newcommand*\GlsSetWriteIstHook[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremake\GlsSetWriteIstHook

```

\@gls@writeisthook

```
\newcommand*\@gls@writeisthook{}
```

\GlsSetQuote Allow user to set the makeindex quote character. This is primarily for ngerman users who want to use makeindex's -g option.

```

\ifglxsindy
\newcommand*\GlsSetQuote[1]{\glsglsmakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote[1]{\glsglsmakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote[1]{\edef\@gls@quotechar{\string#1}%

```


If German is in use, set the extra makeindex option so makeglossaries can pick it up.

```
\@ifpackageloaded{tracklang}%
{%
  \IfTrackedLanguage{german}%
  {%
    \def\@gls@extramakeindexopts{-g}%
  }%
}%
{}%
```

Need to redefine \@gls@checkquote

```
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
```

```

#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescquote###1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescbar###1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%

```

```

\noexpand\def\noexpand\@gls@checkescquote####1%
  \expandonce{\csname#1\endcsname}####2\expandonce{\csname#1\endcsname}%
  ####3\noexpand\null{%
    \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
    \noexpand\toks@={####1}%
    \noexpand\ifx\noexpand\null####2\noexpand\null
      \noexpand\ifx\noexpand\null####3\noexpand\null
        \noexpand\edef\noexpand\@gls@checkedmkidx{%
          \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
        \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
        \noexpand\else
          \noexpand\edef\noexpand\@gls@checkedmkidx{%
            \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
            \noexpand\@gls@quotechar\noexpand\string\expandonce{%
              \csname#1\endcsname}\noexpand\@gls@quotechar
              \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                \csname#1\endcsname}\noexpand\@gls@quotechar}%
            \noexpand\def\noexpand\@gls@checkescquote{%
              \noexpand\@gls@checkescquote####3\noexpand\null}%
          \noexpand\fi
        \noexpand\else
          \noexpand\edef\noexpand\@gls@checkedmkidx{%
            \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
            \noexpand\@gls@quotechar\noexpand\string
              \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
          \noexpand\ifx\noexpand\null####3\noexpand\null
            \noexpand\def\noexpand\@gls@checkescquote{%
              \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
              \expandonce{\csname#1\endcsname}\noexpand\null}%
          \noexpand\else
            \noexpand\def\noexpand\@gls@checkescquote{%
              \noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
              ####3\noexpand\null}%
          \noexpand\fi
        \noexpand\fi
      \noexpand\@gls@checkescquote
    }%
  }%
\noexpand\@gls@docheckquotedef
}
\noexpand\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {\Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\noexpand\fi

```

\@gls@extramakeindexopts

```
\newcommand*{\@gls@extramakeindexopts}[1]{}
```

The command \noist will suppress the creation of the .ist file. Obviously

you need to use this command before `\writeist` to have any effect.

```
\noist
  \newcommand{\noist}{%
Update attributes list
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
```

```
\@gls@noist@toolate
  \newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}
```

`\makeglossary` is an internal command that takes an argument indicating the glossary type. This command will create the glossary file required by `makeindex` for the given glossary type, using the extension supplied by the `<out-ext>` parameter used in `\newglossary` (and it will also activate the `\glossary` command, and create the customized `.ist` `makeindex` style file).

Note that you can't use `\makeglossary` for only some of the defined glossaries (with just the base `glossaries` package). You either need to have a `\makeglossaries` for all glossaries or none (otherwise you will end up with a situation where \TeX is trying to write to a non-existent file). The relevant glossary must be defined prior to using `\makeglossary`. `glossaries-extra` allows for a hybrid approach.

`\@makeglossary` Unstarred form of `\ifglossaryexists` is used as `\makeglossary` can't be used with an ignored glossary.

```
\newcommand*{\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
```

Only create a new write if `savewrites=false` otherwise create a token to collect the information.

```
  \ifglssavewrites
    \expandafter\newtoks\csname glo@#1@filetok\endcsname
  \else
    \expandafter\newwrite\csname glo@#1@file\endcsname
    \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
  \fi
  \@gls@renewglossary
  \writeist
}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘#1’ not defined}%
  {New glossaries must be defined before using \string\makeglossaries}%
}%
```

```

    }

\@glsopenfile Open write file associated with the given glossary.
    \newcommand*{\@glsopenfile}[2]{%
        \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
        \PackageInfo{glossaries}{Writing glossary file
            \jobname.\csname @glotype@#2@out\endcsname}%
    }

\@closegls
    \newcommand*{\@closegls}[1]{%
        \closeout\csname glo@#1@file\endcsname
    }

\@gls@automake Unstarred form of \ifglossaryexists is used as \@gls@automake can't be
used with an ignored glossary.
    \ifglxindy
        \newcommand*{\@gls@automake}[1]{%
            \ifglossaryexists{#1}
            {%
                \@closegls{#1}%
                \ifdefstring{\glsorder}{letter}%
                {\def\@gls@order{-M ord/letorder }}%
                {\let\@gls@order\@empty}%
                \gls@xdy@set@lang{#1}%

                \ifdefvoid\gls@codepage
                {\let\@gls@codepage\@empty}%
                {\def\@gls@codepage{-C \gls@codepage\space}}%
                \edef\@gls@dothiswrite{\noexpand\write18{xindy
                    -I xindy
                    \@gls@order
                    -L \@gls@langmod\space
                    -M \gls@istfilebase\space
                    \@gls@codepage
                    -t \jobname.\csuse{@glotype@#1@log}
                    -o \jobname.\csuse{@glotype@#1@in}
                    \jobname.\csuse{@glotype@#1@out}}}%
               }%
                \@gls@dothiswrite
            }%
            {%
                \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
            }%
        }
    \else
        \newcommand*{\@gls@automake}[1]{%
            \ifglossaryexists{#1}
            {%
                \@closegls{#1}%

```

```

\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{\@glotype@#1@log}
-o \jobname.\csuse{\@glotype@#1@in}
\jobname.\csuse{\@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi

```

`\@gls@automake@immediate` Unstarred form of `\ifglossaryexists` is used as `\@gls@automake@immediate` can't be used with an ignored glossary. Note that “out” means the output file from this package's point of view. It will be the input file from the indexing application's point of view.

```

\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{\@glotype@#1@out}}{%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%

```

Set `\@gls@langmod`:

```

\gls@xdy@set@lang{#1}%

\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
\@gls@codepage
-t \jobname.\csuse{\@glotype@#1@log}
-o \jobname.\csuse{\@glotype@#1@in}
\jobname.\csuse{\@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}}

```

```

        doesn't exist. Rerun may be required}}%
    }%
    {%
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
            {%
                \ifdefstring{\glsorder}{letter}%
                {\def\@gls@order{-l }}%
                {\let\@gls@order\@empty}%
                \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
                    -s \istfilename\space
                    -t \jobname.\csuse{@glotype@#1@log}
                    -o \jobname.\csuse{@glotype@#1@in}
                    \jobname.\csuse{@glotype@#1@out}}}%
                }%
                \@gls@dothiswrite
            }%
            {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
                doesn't exist. Rerun may be required}}%
        }%
        {%
            \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
        }%
    }
}
\fi

```

`\warn@nomakeglossaries` Issue warning that `\makeglossaries` hasn't been used.

```
\newcommand*{\@warn@nomakeglossaries}{}

```

Only use this if warning if `\printglossary` has been used without `\makeglossaries`

```
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}

```

`\@gls@automake@types` The list of glossaries to build if automake is on. Provided for glossaries-extra's hybrid option.

```
\newcommand{\@gls@automake@types}{\@glo@types}

```

`\@gls@@automake@immediate`

```

\newcommand{\@gls@@automake@immediate}{%
    \ifnum\gls@automake@nr=2\relax
        \@for\@gls@type:=\@gls@automake@types\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`

```
\ifglssavenumberlist
\edef\@gls@dodedeflistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodedeflistparser
\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{%
\@for\@gls@type:=\@gls@automake@types\do{%
```

```

\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi

```

Check the sort setting:

```

\@glo@check@sortallowed\makeglossaries
}%
}

```

Must occur in the preamble:

```

\@onlypreamble{\makeglossaries}

```

`saries@warn@noprintglossary`

```

\newcommand\@makeglossaries@warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%

```

`\glswrite` The definition of `\glswrite` has now been moved to `\makeglossaries` so that it's only defined if needed.

If `\makeglossaries` hasn't been used, issue a warning. Also issue a warning if neither `\printglossaries` nor `\printglossary` have been used.

```

\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}

```

`\makenoidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```

\newcommand*\makenoidxglossaries{%
\def\glsindexingsetting{noidx}%
\@domakeglossaries
}%

```

Redefine empty glossary warning:

```

\renewcommand{\@gls@noref@warn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%

```

Don't escape makeindex/xindy characters:

```
\let\@gls@checkmkidxchars\@gobble
```

Don't escape locations:

```
\glsesclocationsfalse
```

Write glossary information to aux instead of glossary files

```
\let\@do@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
\renewcommand{\@do@seeglossary}[2]{%  
  \protected@edef\@gls@label{\glsdetoklabel{##1}}%  
  \protected@write\@auxout{}{%  
    \string\@gls@reference  
      {csname glo\@gls@label @type\endcsname}%  
      {\@gls@label}%  
      {%  
        \string\glsseeformat##2}%  
      }%  
  }%  
}%
```

If user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\AtBeginDocument  
{%  
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%  
}%
```

Change warning about no glossaries

```
\def\warn@noprntglossary{%  
  \GlossariesWarningNoLine{No \string\prntnoidxglossary\space  
    or \string\prntnoidxglossaries ^^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\makenoid@glossaries}%
{Either move your definitions to the preamble or use
\string\makeglossaries}%
}%
```

Redefine \glsentrynumberlist

```
\renewcommand*\glsentrynumberlist[1]{%
\letcs{\@gls@loclist}{glo@glstetoklabel{##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@glstetoklabel{##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@glsetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glssseeformat\glssseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glssseeformat##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\glssseeformat\@gls@org@glssseeformat
}%

```

Modify sanitize sort function

```

\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort

```

Check sort option allowed.

```

\@glo@check@sortallowed\makenoidxglossaries

```

Implement nonumberlist

```

\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}

```

Preamble-only command:

```

\@onlypreamble{\makenoidxglossaries}

```

`\glsnumberlistloop`

```

\glsnumberlistloop{<label>}{<handler>}{<xr handler cs>}

```

```

\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}

```

```

\glsnoidxnumberlistloophandler Handler macro for \glsnumberlistloop. (The argument should be in the form
\glsnoidxdisplayloc{\<prefix>}{\<counter>}{\<format>}{\<n>})
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}

\@no@makeglossaries Can't use both \makeglossaries and \makenoidxglossaries
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}

\@gls@noref@warn Warning when no instances of \@gls@reference found.
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}

```

2.14 Writing information to associated files

```

\gls@noidxglossary Write the glossary information to the aux file (for the 'noidx' method).
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@gls@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@gls@locref}}%
    }%
  }%
}

```

`\istfile` Deprecated in v3.0 (2011-04-02). Removed in v4.50.

At the end of the document, the files should be created if `savewrites=true`.

```

\AtEndDocument{%
  \glswritefiles
}

```

`\@glswritefiles` Only write the files if `savewrites=true`.

```

\newcommand*{\@glswritefiles}{%
  Iterate through all the glossaries.
  \forallglossaries{\@gls@type}{%

```

Check for empty glossaries (patch provided by Patrick Häcker)

```

\ifcsundef{glo@\@glo@type @filetok}%
{%
  \def\gls@tmp{}%
}%
{%
  \edef\gls@tmp{\expandafter\the
    \csname glo@\@glo@type @filetok\endcsname}%
}%
\ifx\gls@tmp\@empty
  \ifx\@glo@type\glsdefaulttype
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries.^^JRemember to use package option ‘nomain’ if
you
      don’t want to^^Juse the main glossary}%
  \else
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries}%
  \fi
\else
  \@glsopenfile{\glswrite}{\@glo@type}%
  \immediate\write\glswrite{%
    \expandafter\the
    \csname glo@\@glo@type @filetok\endcsname}%
  \immediate\closeout\glswrite
  \fi
}%
}

```

As from v4.10, the `\glossary` command isn’t used by the `glossaries` package. Since the user isn’t expected to use this command (as `glossaries` takes care of the particular format required for `makeindex/xindy`) there’s no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```

\gls@glossary
  \newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
  }

```

`\@gls@glossary{<type>}{<indexing info>}`

`\@gls@glossary`

(In v4.10, `\@glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%@\%}`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the sort key), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glsperscentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```
\newcommand*{\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}
```

This is a convenience command to set `\@gls@glossary`. It's used by `\makeglossary` and then redefined to do nothing, as it only needs to be done once.

`\@gls@renewglossary`

```
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
```

The `\gls@wrglossary` command is defined to have two arguments. The first argument is the glossary type, the second argument is the glossary entry (the format of which is set in `\glslink`).

`\gls@wrglossary`

```
\newcommand*{\gls@wrglossary}[2]{%
  \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]{%
    \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]{%
    \ifglindexonlyfirst
        \ifglused{#1}{#2}%
    \else
        #2%
    \fi
}

\gls@protected@pagefmts List of page formats to be protected against expansion.
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

\gls@disablepagerefexpansion As from v4.50, this will only do the page protection formats for escloca-
tions=true.
\newcommand*{\gls@disablepagerefexpansion}{%
    \glswrglossdisablelocationcmds
    \ifglsclocations
        \@for\@gls@this:=\gls@protected@pagefmts\do
        {%
            \expandafter\let\@gls@this\relax
        }%
    \fi
}

\glswrglossdisablelocationcmds

\newcommand{\glswrglossdisablelocationcmds}{}

\gls@alphpage
\newcommand*{\gls@alphpage}{\@alph\c@page}

```

```

\gls@Alphpage
\newcommand*\gls@Alphpage{\@Alph\c@page}

\gls@numberpage
\newcommand*\gls@numberpage{\number\c@page}

\gls@arabicpage
\newcommand*\gls@arabicpage{\@arabic\c@page}

\gls@romanpage
\newcommand*\gls@romanpage{\romannumeral\c@page}

\gls@Romanpage
\newcommand*\gls@Romanpage{\@Roman\c@page}

```

```
\glsaddprotectedpagefmt{<cs name>}
```

```
\glsaddprotectedpagefmt
```

Added a page format to the list of protected page formats. The argument should be the name (without a backslash) of the command that takes a \TeX register as the argument ($\backslash\langle csname\rangle\backslash\c@page$ must be valid).

```

\newcommand*\glsaddprotectedpagefmt[1]{%
  \ifglseclocations
    \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}%
}

```

$\backslash\@wrglossarynumberhook$ Hook used by $\backslash\@do@wrglossary$

```
\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{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
```

Find the prefix if `\@glsHlocref` and `\@glslocref` aren't the same.

```
\ifx\@glsHlocref\@glslocref
  \def\@glo@counterprefix{}%
\else
```

The value of the counter isn't important here as it's the prefix that's of interest. (`\c@page` will have the same value in both `\theglentrycounter` and `\theHglentrycounter` at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

```
\@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
```

De-tok label if required.

```
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```
\@do@wrglossary  
}
```

`\ifglswrallowprimitivemods` Conditional to determine whether or not `\@do@esc@wrglossary` should be allowed to temporarily redefine `\the` and `\number`. Version 4.50 switched the default to false.

```
\newif\ifglswrallowprimitivemods  
\glswrallowprimitivemodsfalse
```

`\@do@esc@wrglossary` Write the glossary entry in the appropriate format. (Need to set `\@glsnumberformat` and `\@gls@counter` prior to use.) The argument is the entry's label. This is far more complicated with `xindy` than with other indexing methods. There are two necessary but conflicting requirements with `xindy`:

1. all backslashes in the location must be escaped;
2. `\c@page` can't be prematurely expanded.

(With `makeindex` there's the remote possibility that the page compositor is a `makeindex` special character, so that would also need to be escaped.)

For example, suppose `\thepage` is defined as

```
%\renewcommand{\thepage}{\tally{page}}  
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}  
%
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won't accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character "t". The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn't be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn't be escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}  
%
```

(note the use of `\expandafter` and `\name`) or in the form

```
%\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}%
%
```

In the second case, `\arabic` is one of the known commands that's temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it's followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH<counter>` is defined in the form `<prefix>.\the<counter>`. It's possible (although again unlikely) that a `makeindex` character might occur in the prefix, which therefore needs escaping. The prefix is passed as the optional argument of `\setentrycounter` which is needed by commands like `\glshypernumber` to create a hyperlink for a given counter (like `\hyperpage` but for an arbitrary counter).

```
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
```

Redefine:

```
\ifglswrallowprimitivemods
```

The redefinition of `\the` to use `\expandafter` solves the problem of `\the\csname c@<counter>\endcsname` but is only a partial solution to the problem of `\the\value`. With `\value`, `\c@page` is too deeply hidden and will be expanded too soon, but at least there won't be an error.

```
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
```

```

\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
  \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%

```

Add hook to allow for other number formats:

```
\@wrglossarynumberhook
```

Prevent expansion:

```
\gls@disablepagerefexpansion
```

Now store location in \@glslocref:

```

\protected@xdef\@glslocref{\theHglentrycounter}%
\endgroup

```

Escape any special characters. It's possible that with `makeindex` the separator might be a `makeindex` special character. Although not likely, it still needs to be taken into account.

```
\@gls@checkmkidxchars\@glslocref
```

Check if the hyper-location is the same as the location and set the hyper prefix.

```

\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@glsHlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi

```

De-tok label if required

```
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```

\@@do@@wrglossary
}

```

```
\@@do@@wrglossary
```

```
\newcommand*{\@@do@@wrglossary}{%
```

Determine whether to use `xindy` or `makeindex` syntax

```
\ifglsxindy
```

Need to determine if the formatting information starts with a (or) indicating a range.

```

\expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
\def\@glo@range{}%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi

```

Write to the glossary file using xindy syntax.

```

\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

:locoref \string"\@glo@counterprefix}{\@gls@locoref}\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@locoref}}%
\fi
}

```

\@xp@gls@getcounterprefix

```

\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%
\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\@do@gls@getcounterprefix
}

```

\@gls@getcounterprefix Get the prefix that needs to be prepended to counter in order to get the hyper counter. (For example, with the standard `article` class and `hyperref`, `\theequation` needs to be prefixed with `\section num` to get the equivalent `\theHequation`.) NB this assumes that the prefix ends with a dot, which is the standard. (Otherwise it makes the xindy location classes more complicated.)

The expansion has been moved to `\@xp@gls@getcounterprefix`.

```

\newcommand*\@gls@getcounterprefix[2]{%

```

```

\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
  \def\@glo@counterprefix{}%
\else
  \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
    \def\@glo@tmp{##2}%
    \ifx\@glo@tmp\@empty
      \def\@glo@counterprefix{}%
    \else
      \def\@glo@counterprefix{##1}%
    \fi
  }%
  \@gls@get@counterprefix#2.#1\end@getprefix
Warn if no prefix can be formed.
\ifx\@glo@counterprefix\@empty
  \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
    prefixing^^Jlocation ‘#1’. You need to modify the
    definition of \string\theH\@gls@counter^^Jotherwise you
    will get the warning: "‘name{\@gls@counter.#1}’ has been^^J
    referenced but does not exist"}%
\fi
\fi
}

```

2.15 Glossary Entry Cross-References

`\@do@seeglossary` Write the glossary entry with a cross reference. The first argument is the entry’s label, the second must be in the form `[(tag)]{<list>}`, where *tag* is a tag such as “see” and *list* is a list of labels.

```

\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string@glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
  \fi
}

```


`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{\{#2#3\}}%
  \fi
}

\@@gls@fixbraces
\def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{\[#2\]{#3}}%
}

\glssee \glssee{\langle label \rangle}{\langle cross-ref list \rangle}
\newrobustcmd*{\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{\[#1\]{#3}}%
  \newcommand*{\@glssee}[3][\seename]{%
    \glssee[#1]{#3}{#2}}
}

\glsseeformat The first argument specifies what tag to use (e.g. “see”), the second argument is
a comma-separated list of labels. The final argument (the location) is ignored.
\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}

\glsseelist \glsseelist{\langle list \rangle} formats list of entry labels.
\newrobustcmd*{\glsseelist}[1]{%
  If there is only one item in the list, set the last separator to do nothing.
  \let\@gls@dolast\relax
  Don't display separator on the first iteration of the loop
  \let\@gls@donext\relax
  Iterate through the labels
  \@for\@gls@thislabel:=#1\do{%
    Check if on last iteration of loop
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    Display the entry for this label. (Expanding label as it's a temporary control
    sequence that's used elsewhere.)
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    Update separators
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}

```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```
\newcommand*{\glsseelastsep}{\space\andname\space}
```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```
\newcommand*{\glsseesep}{, }
```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```
\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
```

`\glsseeitemformat` Introduced in v3.0 to replace use of `\glsentryname` (originally in `\glsseeitem`) with `\glsentrytext` to avoid problems with the name key being sanitized. This ceased being a problem in v3.08a, but `\glsentrytext` was retained for backward-compatibility. As from v4.50, the default has reverted back to using `\glsentryname` unless the entry is an acronym (which could have a potentially long name).

```
\newcommand*{\glsseeitemformat}[1]{%
  \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@\glscurrententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}{\@do@writeaux@info}%
  \fi
}
```

`\warn@noprintglossary` Warn the user if they have forgotten `\printglossaries` or `\printglossary`. (Will be suppressed if there is at least one occurrence of `\printglossary`. There is no check to ensure that there is a `\printglossary` for each defined glossary.)

```
\newcommand*{\warn@noprintglossary}{}%
```

`\printglossary` The TOC title needs to be processed in a different manner to the main title in case the `translator` and `hyperref` packages are both being used.

```
\ifcsundef{printglossary}{}%
{%
```

If `\printglossary` is already defined, issue a warning and undefine it.

```
\@gls@warnonglossdefined
\undef\printglossary
}
```

Neither `\printglossary` nor `\printnoidxglossary` can work with an ignored glossary (since ignored glossaries normally suppress indexing and there's no associated file for `makeindex/xindy` to process). However `\printunsrtglossary` can be used with an ignored glossary, so provide a command to warn if the glossary doesn't exist or is an ignored glossary.

`gloss@checkexists@noignored`

```
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
\@ifglossaryexists{#1}%
{#2}%
{%
\ifignoredglossary{#1}%
{\GlossariesWarning{Glossary '#1' is an ignored glossary}}%
{\GlossariesWarning{Glossary '#1' doesn't exist}}%
}%
}
```

`ss@checkexists@allowignored` For use with `\printunsrtglossary`.

```
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
\s@ifglossaryexists{#1}%
{#2}%
{\GlossariesWarning{Glossary '#1' doesn't exist}}%
}
```

`\@printgloss@checkexists`

```
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
```

`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```
\newcommand*{\printglossary}[1][type=glndefaulttype]{%
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\@printglossary{#1}{\@print@glossary}%
}
```

The `\printglossaries` command will do `\printglossary` for each glossary type that has been defined. It is better to use `\printglossaries` rather than individual `\printglossary` commands to ensure that you don't forget any new glossaries you may have created. It also makes it easier to chop and change the value of the `acronym` package option. However, if you want to list the glossaries

in a different order, or if you want to set the title or table of contents entry, or if you want to use different glossary styles for each glossary, you will need to use `\printglossary` explicitly for each glossary type.

```
\printglossaries
    \newcommand*{\printglossaries}{%
      \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
    }

\printnoidxglossary Provide an alternative to \printglossary that doesn't require an external in-
dexing application. Entries won't be sorted and the location list will be empty.
    \newcommand*{\printnoidxglossary}[1][type=glndefaulttype]{%
      \let\@printgloss@checkexists\@printgloss@checkexists@noignored
      \@printglossary{#1}{\printnoidx@glossary}%
    }

\printnoidxglossaries Analogous to \printglossaries
    \newcommand*{\printnoidxglossaries}{%
      \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
    }

\@printgloss@setsort Initialise to do nothing.
    \newcommand*{\@printgloss@setsort}{}

\@gls@preglossaryhook
    \newcommand*{\@gls@preglossaryhook}{}

\@printglossary Sets up the glossary for either \printglossary or \printnoidxglossary. The
first argument is the options list, the second argument is the handler macro
that deals with the actual glossary. This is also used by glossaries-extra's
\printunsrtglossary which may be used with an ignored glossary.
    \newcommand{\@printglossary}[2]{%
Set up defaults.
      \def\@glo@type{\glndefaulttype}%
      \def\glossarytitle{\csname @glo@type \@glo@type @title\endcsname}%

      \def\glossarytoctitle{\glossarytitle}%
      \let\org@glossarytitle\glossarytitle

      \def\@glossarystyle{%
        \ifx\@glossary@default@style\relax
          \GlossariesWarning{No default glossary style provided \MessageBreak
for the glossary '\@glo@type'. \MessageBreak
Using fallback. \MessageBreak
To fix this set the style with \MessageBreak
\string\setglossarystyle\space or use the \MessageBreak
style key=value option}%
        \fi
      }%
      \def\gls@dotoc@title{\gls@settoc@title{\@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@dotoctitle
```

Set the glossary style

```
\@glossarystyle
```

Added a way to fetch the current entry label (v3.08 updated for new `\glossentry` and `\subglossentry`, but this is now only needed for backward compatibility):

```
\let\gls@org@glossaryentryfield\glossentry  
\let\gls@org@glossarysubentryfield\subglossentry  
\renewcommand{\glossentry}[1]{%  
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%  
\gls@org@glossaryentryfield{##1}%  
}%  
\renewcommand{\subglossentry}[2]{%
```

```

\protected@xdef\glscurrententrylabel{\glsetoklabel{##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@noprintglossary\relax
}

```

`\@print@glossary` Internal workings of `\printglossary` dealing with reading the external file.

```

\newcommand{\@print@glossary}{%

```

Some macros may end up being expanded into internals in the glossary, so need to make `@` a letter. (Unlikely to be a problem since v3.08a but kept for backward compatibility.)

```

\makeatletter

```

Input the glossary file, if it exists.

```

\@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%

```

If the glossary file doesn't exist, do `\null`. (This ensures that the page is shipped out and all write commands are done.) This might produce an empty page, but at this point the document isn't complete, so it shouldn't matter.

```

\IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
{}%
{\null}%

```

If `xindy` is being used, need to write the language dependent information to the `.aux` file for `makeglossaries`.

```

\ifglsxindy

```

Set `\@gls@langmod`:

```

\gls@xdy@set@lang{\@glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%

```

If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%

```

```

\string\@xdylanguage{\@glo@type}{\@gls@langmod}}}%
}%
}%
\do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%

```

If the user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@gls@codepage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@gls@codepage{\@glo@type}{\@gls@codepage}}}%
}%
}%
\do@auxoutstuff
\fi

```

Activate warning if `\makeglossaries` hasn't been used.

```

\renewcommand*{\@warn@nomakeglossaries}{%
\GlossariesWarningNoLine{\string\makeglossaries\space
hasn't been used,^^Jthe glossaries will not be updated}%
}%
}

```

The sort macros all have the syntax:

```
\@glo@sortmacro@<order>{<type>}
```

where `<order>` is the sort order as specified by the sort key and `<type>` is the glossary type. (The referenced entry list is stored in `\@glsref@<type>`). The actual sorting is done by `\@glo@sortentries{<handler>}{<type>}`.

`\@glo@sortentries`

```

\newcommand*{\@glo@sortentries}[2]{%
\glosortentrieswarning
\def\@glo@sortinglist{}%
\def\@glo@sortinghandler{#1}%

\protected@edef\@glo@type{#2}%
\forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
\csdef{\@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%

```

Has this entry already been added?

```

\xifinlistcs{\@this@label}{\@glsref@#2}%
}%
{%
\listcsxadd{\@glsref@#2}{\@this@label}%

```

```

}%
\ifcsdef{@glo@sortingchildren@\@this@label}%
{%
  \@glo@addchildren{#2}{\@this@label}%
}%
{}%
}%
}

```

```
\@glo@addchildren{<type>}{<parent>}
```

\@glo@addchildren

```
\newcommand*{\@glo@addchildren}[2]{%
```

Scope to allow nesting.

```

\bgroup
\letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
\@for\@this@childlabel:=\@glo@childlist\do
{%

```

Check this label hasn't already been added.

```

\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@glsetoklabel{#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?
\ifinlistcs{\@glo@parent}{\@glsref@\@glo@type}%
{%
Yes, it has so do nothing.
}%
{%
No, it hasn't so add it now.
\expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}%
}%
{%
\@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}

```

```
\@glo@sortedinsert{<list>}{<entry label>}
```

\@glo@sortedinsert

Insert into list.

```

\newcommand*{\@glo@sortedinsert}[2]{%
\dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%

```

The sort handlers need to be in the form required by datatool's \dtl@sortlist macro. These must set the count register \dtl@sortresult to either -1 (#1 less than #2), 0 (#1 = #2) or +1 (#1 greater than #2).

\@glo@sorthandler@word

```

\newcommand*{\@glo@sorthandler@word}[2]{%
\letcs\@gls@sort@A{\@glo@\@glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{\@glo@\@glsdetoklabel{#2}@sort}%
\edef\@glo@do@compare{%
\noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
{\expandonce\@gls@sort@B}%
{\expandonce\@gls@sort@A}%
}%
\@glo@do@compare
}

```

\@glo@sorthandler@letter

```

\newcommand*{\@glo@sorthandler@letter}[2]{%
\letcs\@gls@sort@A{\@glo@\@glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{\@glo@\@glsdetoklabel{#2}@sort}%
\edef\@glo@do@compare{%
\noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
{\expandonce\@gls@sort@B}%
}

```

```

        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}

\@glo@sorthandler@case Case-sensitive sort.
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}

\@glo@sorthandler@nocase Case-insensitive sort.
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}

\@glo@sortmacro@word Sort macro for ‘word’
\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{ }%
  }%
}

\@glo@sortmacro@letter Sort macro for ‘letter’
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%

```

```

        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=letter]}{}}%
    }%
}

\@glo@sortmacro@standard Sort macro for ‘standard’. (Use either ‘word’ or ‘letter’ order.)
\newcommand*{\@glo@sortmacro@standard}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \ifcsdef{\@glo@sorthandler@\glsorder}%
        {%
            \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
        }%
        {%
            \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}}%
        }%
    }%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=standard]}{}}%
}

\@glo@sortmacro@case Sort macro for ‘case’
\newcommand*{\@glo@sortmacro@case}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@case}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=case]}{}}%
    }%
}

\@glo@sortmacro@nocase Sort macro for ‘nocase’
\newcommand*{\@glo@sortmacro@nocase}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=nocase]}{}}%
    }%
}

```

`\@glo@sortmacro@def` Sort macro for ‘def’. The order of definition is given in `\glo@list@{type}`.

```
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forl@sentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
  }%
}
```

Hasn’t been referenced.

```
}%
}%
\cslet{\@glsref@#1}{\@glo@sortinglist}%
}
```

`\@glo@sortmacro@def@do` This won’t include parent entries that haven’t been referenced.

```
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
```

`\@glo@sortmacro@use` Sort macro for ‘use’. (No sorting is required, as the entries are already in order of use, so do nothing.)

```
\newcommand*{\@glo@sortmacro@use}[1]{}
```

`\@print@noidx@glossary` Glossary handler for `\printnoidxglossary` which doesn’t use an indexing application. Since `\printnoidxglossary` may occur at the start of the document, we can’t just check if an entry has been used. Instead, the first pass needs to write information to the aux file every time an entry is referenced. This needs to be read in on the second run and stored in a list corresponding to the appropriate glossary.

```
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%

```

Sort the entries:

```
\ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
{%
  \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
}
```

```

}%
{%
    \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
}%

```

Do the glossary heading and preamble

```

\glossarysection[\glossarytoctitle]{\glossarytitle}%
\glossarypreamble

```

The glossary style might use a tabular-like environment, which may cause scoping problems when setting the current letter group. The predefined tabular-like styles don’t support letter group headings, but there’s nothing to stop the user from defining their own custom style that might, so any redefinition of this command within theglossary will have to be done globally.

```

\def\@gls@currentlettergroup{}%
\begin{theglossary}%
\glossaryheader
\glsresetentrylist

```

Iterate through the entries.

```

\forlistcsloop{\@gls@noidx@do}{\@glsref@{\@glo@type}}%

```

Finally end the glossary and do the postamble:

```

\end{theglossary}%
\glossarypostamble
}%
{%
    \@gls@noref@warn{\@glo@type}%
}%
}

```

\glo@grabfirst This won’t work for UTF-8 and inputenc.

```

\def\glo@grabfirst#1#2\@nil{%
    \def\@gls@firsttok{#1}%
    \ifdefempty\@gls@firsttok
    {%
        \def\@glo@thislettergrp{0}%
    }%
    {%

```

Sanitize it:

```

\@onelevel@sanitize\@gls@firsttok

```

Fetch the first letter:

```

\expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
}%
}

```

\@glo@grabfirst

```

\def\@glo@grabfirst#1#2\@nil{%
    \ifdefempty\@glo@thislettergrp

```

```

{%
  \def\@glo@thislettergrp{glssymbols}%
}%
{%
  \count@=\uccode'#1\relax
  \ifnum\count@=0\relax
    \def\@glo@thislettergrp{glssymbols}%
  \else
    \ifdefstring\@glo@sorttype{case}%
    {%
      \count@='#1\relax
    }%
    {%
    }%
  }%
  \edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}

```

\@gls@noidx@do Handler for list iteration used by \@print@noidx@glossary. The argument is the entry label. This only allows one sublevel.

```
\newcommand{\@gls@noidx@do}[1]{%
```

Get this entry's location list

```
\global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
```

Does this entry have a parent?

```
\ifglshasparent{#1}%
{%
```

Has a parent.

```

\gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
  \subglossentry{\gls@level}{#1}{}%
}%
{%
  \subglossentry{\gls@level}{#1}%
  {%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}%
{%

```

Doesn't have a parent Get this entry's sort key

```
\letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%
```

Fetch the first letter:

```
\expandafter\glo@grabfirst\@gls@sort{}{}\@nil
```

```

\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%

```

Do the group header:

```

\ifdefempty{\@gls@currentlettergroup}{}%
{%

```

The group skip may start a new scope, so make a global assignment.

```

\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%

```

```

\global\let\@gls@currentlettergroup\@glo@thislettergrp

```

Do this entry:

```

\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glsnoidxprenumberlist{#1}%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
}

```

\glsnoidxloclist

```
\glsnoidxloclist{\<list cs>}
```

Display location list.

```

\newcommand*{\glsnoidxloclist}[1]{%
\def\@gls@noidxloclist@sep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxloclisthandler}{#1}%
}

```

\glsnoidxloclisthandler Handler for location list iterator.

```

\newcommand*{\glsnoidxloclisthandler}[1]{%
\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
Same as previous location so skip.
}%
{%
\@gls@noidxloclist@sep

```

```

    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

`\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}{}}{}%
  {}%

```



```

\listcs gadd{glo@\glsdetoklabel{#2}@loc list}{#3}%
}%
}

```

The keys that can be used in the optional argument to `\printglossary` or `\printnoidxglossary` are as follows: The `type` key sets the glossary type.

```
\define@key{printgloss}{type}{\def\@glo@type{#1}}
```

The `title` key sets the title used in the glossary section header. This overrides the title used in `\newglossary`.

```

\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotoc title\relax
}

```

The `toctitle` sets the text used for the relevant entry in the table of contents.

```

\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotoc title\relax
}

```

The `style` key sets the glossary style (but only for the given glossary).

```

\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossarystyle{#1}}%
}%
}

```

The `numberedsection` key determines if this glossary should be in a numbered section.

```

\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%

```

```

\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}

```

The `nopostdot` key has the same effect as the package option of the same name.

```

\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}

```

`\GlsEntryCounterLabelPrefix` Make it easier to redefine the label prefix.

```

\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}

```

The conditionals have been moved inside the appropriate commands to make it easier for the user to redefine them in the preamble and selectively switch the counter display on and off. Previously the helper commands were redefined by the `entrycounter` option, which would counteract any earlier customisation.

The `entrycounter` key is the same as the package option but localised to the current glossary.

```

\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}

```

The `subentrycounter` key is the same as the package option but localised to the current glossary. Note that this doesn't affect the parent/child counter attributes, which occurs if `subentrycounter` and `entrycounter` package options are set to true.

```

\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glssubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}

```

The `nonumberlist` key determines if this glossary should have a number list.

```

\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
\else
\def\glossaryentrynumbers##1{##1}%
\fi}

```

The `sort` key sets the glossary sort handler (`\printnoidxglossary` only).

```

\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}

```

`\@glo@no@assign@sortkey` Issue error if used with `\printglossary`

```

\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}

```

`\@@glo@assign@sortkey` For use with `\printnoidxglossary`

```

\newcommand*{\@@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}

```

`\@glsnonextpages` Suppresses the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if `\glsnonextpages` is place in the entry’s description and 3 column tabular style glossary is used.) `\org@glossaryentrynumbers` needs to be set at the start of each glossary, in the event that `\glossaryentrynumber` is redefined.

```

\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}

```

`\@glsnextpages` Activate the next number list only. Global assignments required as it may not occur in the same level of grouping as the next numberlist. (For example, if `\glsnextpages` is place in the entry’s description and 3 column tabular style glossary is used.) `\org@glossaryentrynumbers` needs to be set at the start of each glossary, in the event that `\glossaryentrynumber` is redefined.

```

\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}

```

`\glsresetentrylist` Resets `\glossaryentrynumbers`

```

\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}

```

`\glsnonextpages` Outside of `\printglossary` this does nothing.

```

\newcommand*{\glsnonextpages}{}

```

`\glsnextpages` Outside of `\printglossary` this does nothing.

```

\newcommand*{\glsnextpages}{}

```

Process `entrycounter` and then `subentrycounter` options (this ensures the 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}}%
    \else
      \gls{#1}%
    \fi
  \fi
}

```

`\glsentrycounterlabel` Defines how to display the glossaryentry counter.

```

\newcommand*{\glsentrycounterlabel}{%

```

```

\ifglentrycounter
  \theglossaryentry.\space
\fi
}

```

`\glssubentrycounterlabel` Defines how to display the glossarysubentry counter.

```

\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

`\glentryitem` Step and display glossaryentry counter, if appropriate.

```

\newcommand*{\glentryitem}[1]{%
  \ifglentrycounter
    \glstepentry{#1}\glentrycounterlabel
  \else
    \glresetsubentrycounter
  \fi
}

```

`\glssubentryitem` Step and display glossarysubentry counter, if appropriate.

```

\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glstepsubentry{#1}\glssubentrycounterlabel
  \fi
}

```

`theglossary` (*env.*) If the `theglossary` environment has already been defined, a warning will be issued. This environment should be redefined by glossary styles.

```

\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}

```

The glossary header is given by `\glossaryheader`. This forms part of the glossary style, and must indicate what should appear immediately after the start of the `theglossary` environment. (For example, if the glossary uses a tabular-like environment, it may be used to set the header row.) Note that if you don't want a header row, the glossary style must redefine `\glossaryheader` to do nothing.

```

\glossaryheader
\newcommand*{\glossaryheader}{}

```

`\glstarget`

`\glstarget{<label>}{<name>}`

Provide user interface to `\@glstarget` to make it easier to modify the glossary style in the document.

```
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
```

As from version 3.08, glossary information is now written to the external files using `\glossentry` and `\subglossentry` instead of `\glossaryentryfield` and `\glossarysubentryfield`. The default definition provides backward compatibility for glossary styles that use the old forms.

`\compatibleglossentry`

`\compatibleglossentry{<label>}{<page-list>}`

Removed in v4.50.

`\glossentry`

`\glossentry{<label>}{<page-list>}`

```
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
```

`\glossentryname`

```
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
```

`\Glossentryname`

```
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
```

`\glossentrydesc`

```
\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
```

```

\Glossentrydesc
\newcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}

```

```

\glossentrysymbol
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}

```

```

\Glossentrysymbol
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}

```

```
\compatiblesubglossentry{<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{<level>}{<label>}{<name>}
{<description>}{<symbol>}{<page-list>}
```

\glossarysubentryfield

This command governs how each subentry should be formatted in the glossary. Glossary styles need to redefine this command. Most of the predefined styles ignore *<symbol>*. The first argument is a number indicating the level. (The level should be greater than or equal to 1.)

Removed in v4.50.

`\subglossentry`

```
\subglossentry{<level>}{<label>}{<page-list>}
```

```
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
```

Within each glossary, the entries form distinct groups which are determined by the first character of the `sort` key. When using `makeindex`, there will be a maximum of 28 groups: symbols, numbers, and the 26 alphabetical groups A, ..., Z. If you use `xindy` the groups will depend on whatever alphabet is used. This is determined by the language or custom alphabets can be created in the `xindy` style file. The command `\glsgroupskip` specifies what to do between glossary groups. Glossary styles must redefine this command. (Note that `\glsgroupskip` only occurs between groups, not at the start or end of the glossary.)

`\glsgroupskip`

```
\newcommand*{\glsgroupskip}{}%
```

Each of the 28 glossary groups described above is preceded by a group heading. This is formatted by the command `\glsgroupheading` which takes one argument which is the *label* assigned to that group (not the title). The corresponding labels are: `glssymbols`, `glsnumbers`, A, ..., Z. Glossary styles must 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 `\glsgrouptitle` and `\glsgrouplabel` so that the label is translated into the required title (and vice-versa).

```
\glsgrouptitle{<label>}
```

This command produces the title for the glossary group whose label is given by `<label>`. By default, the group labelled `glssymbols` produces `\glssymbolsgroupname`, the group labelled `glsnumbers` produces `\glsnumbersgroupname` and all the other groups simply produce their label. As mentioned above, the group labels

are: `glssymbols`, `glsnumbers`, A, ..., Z. If you want to redefine the group titles, you will need to redefine this command. Languages other than English may produce labels that are non-expandable, so we need to check for that otherwise it will create a “missing `\endcsname` inserted” error.

```
\glsgetgrouptitle
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
```

`\@gls@getgrouptitle` Gets the group title specified by the label (first argument) and stores in the second argument, which must be a control sequence.

```
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
  Even if the argument appears to be a single letter, it won't be considered a
  single letter by \dtl@ifsingle if it's an active character.
  \def#2{#1}%
  \regex_match:nnT { ^\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{\letcs#2{#1groupname}} }
}
\ExplSyntaxOff
```

`\@gls@noidx@getgrouptitle` Version for the no-indexing app option:

```
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
```

`\glsgetgrouplabel{<title>}`

This command does the reverse to the previous command. The argument is the group title, and it produces the group label. Note that if you redefine `\glsgetgrouptitle`, you will also need to redefine `\glsgetgrouplabel`.

`\glsgetgrouplabel` This only works for the symbol and number groups, so is best avoided and not documented. However, it has been converted to use L^AT_EX3 commands.

```
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { glsnumbers }
    { #1 }
  }
}
```

```

}
}
\ExplSyntaxOff

```

The command `\setentrycounter` sets the entry’s associated counter (required by `\glshypernumber` etc.) `\glslink` and `\glsadd` encode the `\glossary` argument so that the relevant counter is set prior to the formatting command.

`\setentrycounter` The prefix is an optional argument because this command originally only had one argument. The optional argument was added in v3.0.

```

\newcommand*{\setentrycounter}[2][{}]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}

```

The current glossary style can be set using `\setglossarystyle{<style>}`.

`\setglossarystyle`

```

\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

Set the default style if it’s not already set.

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}

```

`\glossarystyle` Deprecated in v3.08a. Removed in v4.50.

`\newglossarystyle` New glossary styles can be defined using:

```

\newglossarystyle{<name>}{<definition>}

```

The `<definition>` argument should redefine `\theglossary`, `\glossaryheader`, `\glsgroupheading`, `\glossaryentryfield` and `\glsgroupskip` (see [subsection 2.19](#) for the definitions of predefined styles). Glossary styles should not redefine `\glossarypreamble` and `\glossarypostamble`, as the user should be able to switch between styles without affecting the pre- and postambles.

```

\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}

```

`\renewglossarystyle` Code for this macro supplied by Marco Daniel.

```

\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}

```

Glossary entries are encoded so that the second argument to `\glossaryentryfield` is always specified as `\glsnamefont{<name>}`. This allows the user to change the font used to display the name term without having to redefine `\glossaryentryfield`. The default uses the surrounding font, so in the list type styles (which place the name in the optional argument to `\item`) the name will appear in bold.

`\glsnamefont`

```

\newcommand*{\glsnamefont}[1]{#1}

```

Each glossary entry has an associated number list (usually page numbers) that indicate where in the document the entry has been used. The format for these number lists can be changed using the `format` key in commands like `\glslink`. The default format is given by `\glshypernumber`. This takes a single argument which may be a single number, a number range or a number list. The number ranges are delimited with `\delimR`, the number lists are delimited with `\delimN`.

If the document doesn’t have hyperlinks, the numbers can be displayed just as they are, but if the document supports hyperlinks, the numbers should link to the relevant location. This means extracting the individual numbers from the list or ranges. The `hyperref` package does this with the `\hyperpage` command, but this is encoded for comma and dash delimiters and only for the page counter, but this code needs to be more general. So I have adapted the code used in the `hyperref` package.

`\glshypernumber`

```

\ifcsundef{hyperlink}%
{%

```

```

\def\glshypernumber#1{#1}%
}%
{%
\def\glshypernumber#1{\xp@glshypernumber{#1}}
}

\xp@glshypernumber
\newrobustcmd{\xp@glshypernumber}[1]{%
\@glshypernumber#1\nohyperpage{}\@nil
}

\@glshypernumber This code was provided by Heiko Oberdiek to allow material to be attached to
the location.
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
\ifx\#1\%
\else
\gls@delimR#1\delimR\delimR\%
\fi
\ifx\#2\%
\else
#2%
\fi
\ifx\#3\%
\else
\@glshypernumber#3\@nil
\fi
}

\gls@delimR displays a range of numbers for the counter whose name is given
by \@gls@counter (which must be set prior to using \glshypernumber).

\gls@delimR
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
\gls@delimN{#1}%
\else
\@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}

\gls@delimN displays a list of individual numbers, instead of a range:

\gls@delimN
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
\@gls@numberlink{#1}%
\else
\@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}

```

The following code is modified from hyperref's `\HyInd@pagelink` where the name of the counter being used is given by `\@gls@counter`.

```
\def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil
  \endgroup
}

\def\@gls@removespaces#1 #2\@nil{%
  \toks@=\expandafter{\the\toks@#1}%
  \ifx\#2\%

    \edef\@gls@x{\the\toks@}%
    \ifx\@gls@x\empty
    \else

      \bgroup
```

Allow any markup to be stripped from the location when forming the target name.

```
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glstentrycounter\@glo@counterprefix
  \glswrglosslocationtarget{\@gls@x}}%
```

If anything hasn't been fully expanded by this point then it will cause a problem, so sanitize to prevent an error, but it's likely the target won't be valid.

```
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
  \noexpand\hyperlink{\@gls@tmp}%
  {\noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
```

The following commands will switch to the appropriate font, and create a hyperlink, if hyperlinks are supported. If hyperlinks are not supported, they will just display their argument in the appropriate font.

`\hyperrm`

```
\newcommand*{\hyperrm}[1]{\textrm{\glshypernumber{#1}}}
```

`\hypersf`

```
\newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}
```

```

\hypertt
\newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}

\hyperbf
\newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}

\hypermd
\newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}

\hyperit
\newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}

\hypersl
\newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}

\hyperup
\newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}

\hypersc
\newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}

\hyperemph
\newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}

```

2.17 Acronyms

\oldacronym

```
\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 needed 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}}
```

`\acronymformat` The styles that allow an additional description use `\acronymformat{<short>}{<long>}` to determine what information is displayed in the name.

```
\newcommand*{\acronymformat}[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@acentryname
  Change the way acronyms are defined:
  \renewcommand{\newacronym}[4][{}]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
    }%
  }%
}
```

```

shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%

longplural={\the\glslongtok\noexpand\glspluralsuffix},%
\GenericAcronymFields,%
\the\glskeylisttok
}%
}%
\@do@newglossaryentry
}%

```

Make sure that `\acrfull` etc reflects the new style:

```

\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \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`

```

\acronymsort{<short>}{<long>}

```

Default sort format for acronyms.

```

\newcommand*{\acronymsort}[2]{#1}

```

`\setacronymstyle`

`\setacronymstyle{<style name>}`

```
\newcommand*{\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
  }%
}
```

`\newacronymstyle`

`\newacronymstyle{<style name>}{<entry format
definition>}{<display definitions>}`

Defines a new acronym style called *<style name>*.

```
\newcommand*{\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
```

`\renewacronymstyle` Redefines the given acronym style.

```
\newcommand*{\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
  }%
}
```

`\GlsUseAcrEntryDispStyle`

```
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
```

`\GlsUseAcrStyleDefs`

```
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
```

Predefined acronym styles:

`long-short` $\langle long \rangle$ ($\langle short \rangle$) acronym style.

```
\newacronymstyle{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]{%  
    \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}{\glsngenacfmt}{\glsngenentryfmt}%  
}%  
{%  
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
```

```

\renewcommand*{\genacrfullformat}[2]{%
\glentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`\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.

```

\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})%
\ifdim\dimen@<3em~\else\space\fi
}

```

short-long *<short>* (*<long>*) acronym style.

```

\newacronymstyle{short-long}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}

```



```

}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\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}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}

sm-short-long <short> (\textsmaller{<long>}) acronym style.
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}

```

`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]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }

```

`long-sp-short-desc` $\langle long \rangle$ ($\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply). The space between the long and short form is given by `\glsacspace`.

```

\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##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]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }

```

long-sm-short-desc $\langle long \rangle$ ($\backslash\text{textsmaller}\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

short-long-desc $\langle short \rangle$ ($\{\langle long \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

sc-short-long-desc $\langle long \rangle$ ($\backslash\text{textsc}\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

sm-short-long-desc $\langle long \rangle$ ($\backslash\text{textsmaller}\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
```

```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}

```

dua *<long>* only acronym style.

```

\newacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
  \ifglshaslong{\glslabel}%
  {%
    \glsifplural
    {%

```

Plural form:

```

\glscapscase
{%

```

Plural form, don't adjust case:

```

\glsentrylongpl{\glslabel}\glsinsert
}%
{%

```

Plural form, make first letter upper case:

```

\Glsentrylongpl{\glslabel}\glsinsert
}%
{%

```

Plural form, all caps:

```

\glssupercase
{\glsentrylongpl{\glslabel}\glsinsert}%
}%
{%

```

Singular form

```

\glscapscase
{%

```

Singular form, don't adjust case:

```

\glsentrylong{\glslabel}\glsinsert
}%
{%

```

Subsequent singular form, make first letter upper case:

```

\Glsentrylong{\glslabel}\glsinsert
}%
{%

```

Subsequent singular form, all caps:

```

        \glsupercase
        {\glsentrylong{\glslabel}\glsinsert}%
    }%
}%
}%
{%

```

Not an acronym:

```

        \glsgenentryfmt
    }%
}%
{\glscustomtext\glsinsert}%
}%
{%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

    \renewcommand*{\acrfullfmt}[3]{%
        \glslink[##1]{##2}{\glsentrylong{##2}##3\space
            (\acronymfont{\glsentryshort{##2}})}%
    \renewcommand*{\Acrfullfmt}[3]{%
        \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
            (\acronymfont{\glsentryshort{##2}})}%
    \renewcommand*{\ACRfullfmt}[3]{%
        \glslink[##1]{##2}{%
            \glsupercase{\glsentrylong{##2}##3\space
                (\acronymfont{\glsentryshort{##2}})}%
        }%

    \renewcommand*{\acrfullplfmt}[3]{%
        \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
            (\acronymfont{\glsentryshortpl{##2}})}%
    \renewcommand*{\Acrfullplfmt}[3]{%
        \glslink[##1]{##2}{%
            \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}%
}

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}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
  \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}{%
    \glssupercase{\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}{%
    \glssupercase{\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}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
  {%
    \GlsUseAcrEntryDisplayStyle{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}%
  {%
    \GlsUseAcrEntryDisplayStyle{footnote-sc}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote-sc}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accom-
panying description (which the user needs to supply).
  \newacronymstyle{footnote-sm-desc}%
  {%
    \GlsUseAcrEntryDisplayStyle{footnote-sm}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote-sm}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }

```



```

\DefineAcronymSynonyms
    \newcommand*{\DefineAcronymSynonyms}{%

        Short form

        \acs
            \let\acs\acrshort

        First letter uppercase short form

        \Acs
            \let\Acs\Acrshort

        Plural short form

        \acsp
            \let\acsp\acrshortpl

        First letter uppercase plural short form

        \Acsp
            \let\Acsp\Acrshortpl

        Long form

        \acl
            \let\acl\aclong

        Plural long form

        \aclp
            \let\aclp\aclongpl

        First letter upper case long form

        \Acl
            \let\Acl\Aclong

        First letter upper case plural long form

        \Aclp
            \let\Aclp\Aclongpl

        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.

`\expand@assign@default@plurals` `\@@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@gls@@default@acr@plural`

`\gls@@default@acr@firstplural`

`\gls@@default@acr@descplural`

`\SetDefaultAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\acrfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
```

`\acrlinkfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink{#1}{#2}{#3}}%
}
```

`\acrnolinkfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrnolinkfootnote}[3]{%
  \footnote{#3}%
}
```

`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.

`DescriptionFootnoteNewAcronymDef` 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.

`ShortDescriptionDUAAcronymStyle` Description, don't use acronym and no footnote. Note that the short form is stored in the symbol key, so if the short form needs to be displayed in the glossary, use a style the displays the symbol. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\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 `\acrnameformat` to allow the user to override the way the name is displayed in the list of acronyms. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\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@\glsdetoklabel{#1}@level\endcsname
}
```

`\showglotext`

```
\showglotext{<label>}
```

```
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
```

\showgloplural

\showgloplural{<label>}

```
\newcommand*{\showgloplural}[1]{%
  \expandafter\show\csname glo@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
}
```

\showgloftype

\showgloftype{<label>}

```
\newcommand*{\showgloftype}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
}
```

\showglocounter

\showglocounter{<label>}

```
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
}
```

\showglouser

\showglouser{<label>}

```
\newcommand*{\showglouser}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useri\endcsname
}
```

\showglouserii

\showglouserii{<label>}

```
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userii\endcsname
}
```

\showglouseriii

\showglouseriii{<label>}

```
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriii\endcsname
}
```

\showglouseriv

\showglouseriv{<label>}

```
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\endcsname
}
```

\showglouserv

\showglouserv{<label>}

```
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userv\endcsname
}
```

\showglouservi

\showglouservi{<label>}

```
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@uservi\endcsname
}
```

\showgloname

\showgloname{<label>}

```
\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@name\endcsname
}
```


\showglodesc

\showglodesc{*label*}

```
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
}
```

\showglodescplural

\showglodescplural{*label*}

```
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
}
```

\showglosort

\showglosort{*label*}

```
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
}
```

\showglosymbol

\showglosymbol{*label*}

```
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
}
```

\showglosymbolplural

\showglosymbolplural{*label*}

```
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
}
```

\showgloshort

\showgloshort{*label*}

```
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
}
```

`\showglolong`

`\showglolong{<label>}`

```
\newcommand*{\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
```

`\showgloindex`

`\showgloindex{<label>}`

```
\newcommand*{\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
```

`\showgloflag`

`\showgloflag{<label>}`

```
\newcommand*{\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
```

`\showgloloclist`

`\showgloloclist{<label>}`

```
\newcommand*{\showgloloclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
```

`\showglofield`

`\showglofield{<label>}{<field>}`

```
\newcommand*{\showglofield}[2]{%
  \csshow{glo@\glsdetoklabel{#1}@#2}%
}
```

`\showacronymlists`

`\showacronymlists`

Show list of glossaries that have been flagged as a list of acronyms.

```
\newcommand*{\showacronymlists}{%
  \show\@glsacronymlists
}
```

<code>\showglossaries</code>	<div><code>\showglossaries</code></div> <p>Show list of defined glossaries.</p> <pre>\newcommand*{\showglossaries}{% \show\@glo@types }</pre>
<code>\showglossaryin</code>	<div><code>\showglossaryin{<glossary-label>}</code></div> <p>Show the ‘in’ extension for the given glossary.</p> <pre>\newcommand*{\showglossaryin}[1]{% \expandafter\show\csname @glotype@#1in\endcsname }</pre>
<code>\showglossaryout</code>	<div><code>\showglossaryout{<glossary-label>}</code></div> <p>Show the ‘out’ extension for the given glossary.</p> <pre>\newcommand*{\showglossaryout}[1]{% \expandafter\show\csname @glotype@#1out\endcsname }</pre>
<code>\showglossarytitle</code>	<div><code>\showglossarytitle{<glossary-label>}</code></div> <p>Show the title for the given glossary.</p> <pre>\newcommand*{\showglossarytitle}[1]{% \expandafter\show\csname @glotype@#1title\endcsname }</pre>
<code>\showglossarycounter</code>	<div><code>\showglossarycounter{<glossary-label>}</code></div> <p>Show the counter for the given glossary.</p> <pre>\newcommand*{\showglossarycounter}[1]{% \expandafter\show\csname @glotype@#1counter\endcsname }</pre>
<code>\showglossaryentries</code>	<div><code>\showglossaryentries{<glossary-label>}</code></div> <p>Show the list of entry labels for the given glossary.</p> <pre>\newcommand*{\showglossaryentries}[1]{% \expandafter\show\csname glolist@#1\endcsname }</pre>

2.21 Compatibility with version 2.07 and below

The backward-compatibility options have been removed in version 4.50. Use the rollback feature instead if backward-compatibility is required. Bear in mind that v2.07 was released 2010-07-10 and v3.07 was released 2013-07-05. Any old documents from that long ago will probably have problems with recent kernels.

In order to fix some bugs in v3.0, it was necessary to change the way information is written to the `glo` file, which also meant a change in the format of the Xindy style file. The compatibility option is meant for documents that use a customised Xindy style file with `\noist`. With the compatibility option, hopefully xindy will still be able to process the old document, but the bugs will remain. The issues in versions 2.07 and below:

- With xindy, the counter used by the entry was hard-coded into the Xindy style file. This meant that you couldn't use the `counter` to swap counters.
- With both xindy and makeindex, if used with `hyperref` and `\theH{counter}` was different to `\thecounter`, the link in the location number would be undefined.

3 Glossary Styles

3.1 Glossary hyper-navigation definitions (glossary-hypernav.sty)

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-hypernav-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-hypernav-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2022/10/14 v4.50 (NLCT)]
```

The commands defined in this package are provided to help navigate around the groups within a glossary (see [subsection 2.16](#).) `\printglossary` (and `\printglossaries`) set `\@glo@type` to the label of the current glossary. This is used to create a unique hypertarget in the event of multiple glossaries.

```
\glsnavhyperlink[<type>]{<label>}{<text>}
```

This command makes `<text>` a hyperlink to the glossary group whose label is given by `<label>` for the glossary given by `<type>`.

`\glsnavhyperlink` The optional argument is the glossary type and can be used to make the group label unique in case the same group occurs in multiple glossaries. The optional argument was `\@glo@type` but is now `\currentglossary` because that's less likely to change within the glossary. It's unlikely that the optional argument will be used to set it to anything other than `\currentglossary` but it makes it possible to hyperlink to the group from outside of the glossary.

```
\newcommand*{\glsnavhyperlink}[3][\currentglossary]{%
```

```

\protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
\@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}

```

`\glsnavhyperlinkname` Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
```

```
\glsnavhypertarget[<type>]{<label>}{<text>}
```

`\glsnavhypertarget`

This command makes `<text>` a hypertarget for the glossary group whose label is given by `<label>` in the glossary given by `<type>`. If `<type>` is omitted, `\gls@type` is used which is set by `\printglossary` to the current glossary label. The optional argument complicates things for `glossaries-extra`.

```
\newcommand*{\glsnavhypertarget}{\protect\@gls@navhypertarget}
```

`\@gls@navhypertarget`

```

\newcommand*{\@gls@navhypertarget}[3][\currentglossary]{%
\@glsnavhypertarget{#1}{#2}{#3}%
}

```

The actual code is now in an internal command that doesn't have an optional argument, which makes it easier to save and restore the original behaviour.

`\@glsnavhypertarget`

```
\newcommand*{\@glsnavhypertarget}[3]{%
```

Add this group to the aux file for re-run check.

```
\protected@write\@auxout{}{\string\gls@hypergroup{#1}{#2}}%
```

Add the target.

```
\@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
```

Check list of known groups to determine if a re-run is required.

```

\expandafter\let
\expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname

```

Iterate through list and terminate loop if this group is found.

```

\@for\@gls@elem:=\@gls@list\do{%
\ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
}

```

Check if list terminated prematurely.

```

\if@endfor
\else

```

This group was not included in the list, so issue a warning.

```

\GlossariesWarningNoLine{Navigation panel
for glossary type '#1'~Jmissing group '#2'}%
\gdef\gls@hypergroup@prerun{%
\GlossariesWarningNoLine{Navigation panel

```

```

        has changed. Rerun LaTeX}}%
    \fi
}

```

`\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]{%
  \ifcsdef{@gls@hypergrouplist@#1}%
  {\protected@csxappto{@gls@hypergrouplist@#1}{, #2}}%
  {\protected@csxdef{@gls@hypergrouplist@#1}{#2}}%
}

```

The `\glsnavigation` command displays a simple glossary group navigation. The symbol and number elements are defined separately, so that they can be suppressed if need be. (In earlier versions this command will produce a link to all 28 groups, but some groups may not be defined if there are groups that do not contain any terms, in which case you will get an undefined hyperlink warning. Version 1.14 changed this to only use labels for groups that are present.) Now for the whole navigation bit:

`\glsnavigation`

```

\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergrouplist@\currentglossary}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \letcs\@gls@list{@gls@hypergrouplist@\currentglossary}%
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between

    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \expandafter\glsnavhyperlink\expandafter{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glshypernavsep
  }%
}

```

`\glshypernavsep` Separator for the hyper navigation bar.

```

\newcommand*{\glshypernavsep}{\space\textbar\space}

```

The `\glssymbolnav` produces a simple navigation set of links for just the symbol and number groups. This used to be used at the start of `\glsnavigation`. This command is no longer needed.

```
\glssymbolnav
\newcommand*{\glssymbolnav}{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}
```

3.2 In-line Style (glossary-inline.sty)

This defines an in-line style where the entries are comma-separated with just the name and description displayed.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-inline-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-inline-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-inline}[2022/10/14 v4.50 (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}%
```

```

\glsinlineformat{##1}{%
  \glossentryname{##1}%
}%
\ifglshasdesc{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\glsinlineifhaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%

```

Sub-entries display description:

```

\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glsresetsubentrycounter
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%

```

Nothing special between groups:

```

\renewcommand*{\glsgroupskip}{}%
}

```

`\glsinlineifhaschildren`

```

\newcommand{\glsinlineifhaschildren}[3]{%
  \ifglshaschildren{#1}{#2}{#3}%
}

```

`\glsinlinedopostchild`

```

\newcommand*{\glsinlinedopostchild}{%

```



```

\gls@inlinepostchild
\def\gls@inlinepostchild{%
}

\glsinlineseparator Separator to use between entries.
\newcommand*\glsinlineseparator{;\space}

\glsinlinesubseparator Separator to use between sub-entries.
\newcommand*\glsinlinesubseparator{,\space}

\glsinlineparentchildseparator Separator to use between parent and children.
\newcommand*\glsinlineparentchildseparator{: \space}

\glsinlinepostchild Hook to use between child and next entry
\newcommand*\glsinlinepostchild{}

\glspostinline Terminator for inline glossary.
\newcommand*\glspostinline{\glspostdescription\space}

\glsinlinenameformat Formats the name of the entry (first argument label, second argument name):
\newcommand*\glsinlinenameformat[2]{\glstarget{#1}{#2}}

\glsinlinedescformat Formats the entry's description, symbol and location list:
\newcommand*\glsinlinedescformat[3]{\space#1}

\glsinlineemptydescformat Formats the entry's symbol and location list when the description is empty:
\newcommand*\glsinlineemptydescformat[2]{\space#1}

\glsinlinesubnameformat Formats the name of the subentry (first argument label, second argument name):
\newcommand*\glsinlinesubnameformat[2]{\glstarget{#1}{#2}}

\glsinlinesubdescformat Formats the subentry's description, symbol and location list:
\newcommand*\glsinlinesubdescformat[3]{\space#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{v4.50}{2022-10-14}`

Declare package:

`\ProvidesPackage{glossary-list}[2022/10/14 v4.50 (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 hypertext:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt  
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]}
```

altlist The altlist glossary style is like the list style, but places the description on a new line. Sub-entries follow in separate paragraphs without the sub-entry name. This style does not use the entry's symbol.

```
\newglossarystyle{altlist}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Main (level 0) entries start a new item in the list with a line break after the entry name:

```
\renewcommand*{\glossentry}[2]{%
\item[\glsentryitem{##1}%
\glstarget{##1}{\glossentryname{##1}}]%
```

Version 3.04 changed `\newline` to the following paragraph break stuff (thanks to Daniel Gebhardt for supplying the fix) to prevent a page break occurring at this point.

```
\mbox{} \par \nobreak \@afterheading
\glossentrydesc{##1} \glspostdescription \space ##2}%
```

Sub-entries start a new paragraph:

```
\renewcommand{\subglossentry}[3]{%
\par
\glssubentryitem{##2}%
\glstarget{##2}{\strut} \glossentrydesc{##2} \glspostdescription \space ##3}%
}
```

altlistgroup The altlistgroup glossary style is like the altlist style, but the glossary groups have headings.

```
\newglossarystyle{altlistgroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
```

altlisthypergroup The altlisthypergroup glossary style is like the altlistgroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{altlisthypergroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%
\glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertext:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt
{\glshypertarget{##1}{\glsgrouptitle{##1}}}]}
```

`listdotted` The `listdotted` glossary style was supplied by Axel Menzel. I've modified it slightly so that the distance from the start of the name to the end of the dotted line is specified by `\glslistdottedwidth`. Note that this style ignores the page numbers as well as the symbol. Sub-entries are displayed in the same way as top-level entries.

```
\newglossarystyle{listdotted}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Each main (level 0) entry starts a new item:

```
\renewcommand*{\glossentry}[2]{%
  \item[\makebox[\glslistdottedwidth][l]{%
    \glsentryitem{##1}%
    \glstarget{##1}{\glossentryname{##1}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
```

Sub entries have the same format as main entries:

```
\renewcommand*{\subglossentry}[3]{%
  \item[\makebox[\glslistdottedwidth][l]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\glossentryname{##2}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
}
```

```
\glslistdottedwidth
```

```
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
```

`sublistdotted` This style is similar to the `glostylelistdotted` style, except that the main entries just have the name displayed.

```
\newglossarystyle{sublistdotted}{%
```

Base it on the `listdotted` style:

```
\setglossarystyle{listdotted}%
```

Main (level 0) entries just display the name:

```
\renewcommand*{\glossentry}[2]{%
  \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}}%
}
```

3.4 Glossary Styles using `longtable` (`glossary-long.sty`)

The glossary styles defined in the `glossary-long` package used the `longtable` environment in the glossary.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-long-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-long-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2022/10/14 v4.50 (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*{\glsgroupeheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%  
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline  
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

longborder The `longborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
\begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

longheader The `longheader` style is like the `long` style but with a header:

```
\newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
\bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
```

longheaderborder The `longheaderborder` style is like the `long` style but with a header and border:

```
\newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
\setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%
\hline\bfseries \entryname & \bfseries
\descriptionname\tabularnewline\hline
\endhead
\hline\endfoot}%
}
```

long3col The `long3col` style is like `long` but with 3 columns

```
\newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glstdescwidth}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{\glstdescwidth}|p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```


long3colheader The long3colheader style is like long3col but with a header row:

```
\newglossarystyle{long3colheader}{%
Base it on the glosstylelong3col style:
\setglossarystyle{long3col}%
Set the table's header:
\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\endhead}%
}
```

long3colheaderborder The long3colheaderborder style is like the above but with a border

```
\newglossarystyle{long3colheaderborder}{%
Base it on the glosstylelong3colborder style:
\setglossarystyle{long3colborder}%
Set the table's header and add horizontal line at table's foot:
\renewcommand*{\glossaryheader}{%
\hline
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
```

long4col The long4col style has four columns where the third column contains the value of the associated symbol key.

```
\newglossarystyle{long4col}{%
Use a longtable with 4 columns:
\renewenvironment{theglossary}%
{\begin{longtable}{l111}}%
{\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]{%
\glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} &
##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`long4colheader` The `long4colheader` style is like `long4col` but with a header row.

```
\newglossarystyle{long4colheader}{%
Base it on the glostylelong4col style:
\setglossarystyle{long4col}%
Table has a header:
\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries \symbolname&
\bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long4colborder` The `long4colborder` style is like `long4col` but with a border.

```
\newglossarystyle{long4colborder}{%
Base it on the glostylelong4col style:
\setglossarystyle{long4col}%
Use a longtable with 4 columns surrounded by vertical lines:
\renewenvironment{theglossary}%
{\begin{longtable}{|l|l|l|l|}}%
{\end{longtable}}%
Add horizontal lines to the head and foot of the table:
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`long4colheaderborder` The `long4colheaderborder` style is like the above but with a border.

```
\newglossarystyle{long4colheaderborder}{%
Base it on the glostylelong4col style:
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|l|l|l|}}%
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

altlong4col The `altlong4col` style is like the `long4col` style but can have multiline descriptions and page lists.

```
\newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsgdescwidth}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{\glsgdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

altlong4colborder The `altlong4colborder` style is like `altlong4col` but with a border.

```
\newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
\setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
  {\end{longtable}}%
}
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `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{\glstdescwidth}|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{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2022/10/14 v4.50 (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{\glsdscwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

Ragged styles.

longragged-booktabs The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long-booktabs style as a base.

```
\setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{longtable}}%
}
```

longragged3col-booktabs The longragged3col-booktabs style is similar to the longragged3col style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{longragged3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
\setglossarystyle{long3col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

altlongragged4col-booktabs The altlongragged4col-booktabs style is similar to the altlongragged4col style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{altlongragged4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the altlong4col-booktabs style as a base.

```
\setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

```

\glsLTpenaltycheck
\newcommand*\glsLTpenaltycheck}{%
\ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}

```

```

\glspenaltygroupskip
\newcommand{\glspenaltygroupskip}{%
\noalign{\penalty-50\vskip\normalbaselineskip}}

```

`\glsrestoreLToutput` Provide a way of restoring `\LT@output` for the user.

```

\let\@gls@org@LT@output\LT@output
\newcommand*\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}

```

This is David's patch, but I've replaced the hard-coded values with `\glsLTpenaltycheck` to make it easier to adjust.

```

\glspatchLToutput
\newcommand*\glspatchLToutput}{%
\renewcommand*\LT@output}{%
\ifnum\outputpenalty <-\@Mi
\ifnum\outputpenalty > -\LT@end@open
\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{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2022/10/14 v4.50 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the longtable package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`longragged` The longragged glossary style is like the long but uses ragged right formatting for the description column.

```
\newglossarystyle{longragged}{%
```

Use longtable with two columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{glsdescwidth}}}%
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}%
  \glspostdescription\space ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

longraggedborder The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|>{\raggedright}p{\glstdescwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

longraggedheader The `longraggedheader` style is like the `longragged` style but with a header:

```
\newglossarystyle{longraggedheader}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\endhead}%
}
```

`longraggedheaderborder` The `longraggedheaderborder` style is like the `longragged` style but with a header and border:

```
\newglossarystyle{longraggedheaderborder}{%
```

Base it on the `glostylelongraggedborder` style:

```
\setglossarystyle{longraggedborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`longragged3col` The `longragged3col` style is like `longragged` but with 3 columns

```
\newglossarystyle{longragged3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
{ \begin{longtable}{l>{\raggedright}p{\glsgdescwidth}%
  >{\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]{%
  \glsgroupitem{##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>)

```
\ifglsgroupskip
\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{\glsgdescwidth}|%
>{\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{\glsgdescwidth}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]{%
\glstryitem{##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{\glsgdescwidth}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}{|l|>{\raggedright}p{\glsdescwidth}|l|}%
   >{\raggedright}p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`altlongragged4colheaderborder` The `altlongragged4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlongragged4colheaderborder}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}%
   >{\raggedright}p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

3.7 Glossary Styles using multicols (glossary-mcols.sty)

The `glossary-mcols` style file defines glossary styles that use the `multicols` package. These use the tree-like glossary styles in a `multicols` 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{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2022/10/14 v4.50 (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}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%

      \let\item\glstreeitem}%
    {\end{multicols}}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}}%
}
```


`mcoltreegroup` Like the `mcoltree` style but the glossary groups have headings.

```
\newglossarystyle{mcoltreegroup}{%
```

Base it on the `glostylemcoltree` style:

```
\setglossarystyle{mcoltree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
```

`mcoltreehypergroup` The `mcoltreehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcoltreehypergroup}{%
```

Base it on the `glostylemcoltree` style:

```
\setglossarystyle{mcoltree}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreespannav` Similar to the `mcoltreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreespannav}{%
```

```
\setglossarystyle{tree}%
```

```
\renewenvironment{theglossary}%
```

```
{%
```

```
\begin{multicols}{\glsmcols}[ \noindent\glstreenavigationfmt{\glsnavigation}]
```

```
\setlength{\parindent}{0pt}%
```

```
\setlength{\parskip}{0pt plus 0.3pt}%
```

```
}%
```

```
{\end{multicols}}}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreenoname` Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}

```

mcoltreenonamegroup Like the **mcoltreenoname** style but the glossary groups have headings.

```

\newglossarystyle{mcoltreenonamegroup}{%
Base it on the glostylemcoltreenoname style:
  \setglossarystyle{mcoltreenoname}%
Give each group a heading:
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}

```

mcoltreenonamehypergroup The **mcoltreenonamehypergroup** style is like the **mcoltreenonamegroup** style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcoltreenonamehypergroup}{%
Base it on the glostylemcoltreenoname style:
  \setglossarystyle{mcoltreenoname}%
Put navigation links to the groups at the start of the theglossary environment:
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Each group has a heading (in bold with a target) followed by a vertical gap):
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}

```

mcoltreenonamespannav Similar to the **mcoltreenonamehypergroup** style but the navigation line is put in the optional argument of the multicols environment.

```

\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

mcolalttree Multi-column index style. Same as the **alttree**, but puts the glossary in multiple columns.

```
\newglossarystyle{mcolalttree}{%
\setglossarystyle{alttree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
}
```

mcolalttreegroup Like the **mcolalttree** style but the glossary groups have headings.

```
\newglossarystyle{mcolalttreegroup}{%
```

Base it on the **glostylemcolalttree** style:

```
\setglossarystyle{mcolalttree}%
```

Give each group a heading.

```
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
```

mcolalttreehypergroup The **mcolalttreehypergroup** style is like the **mcolalttreegroup** style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcolalttreehypergroup}{%
```

Base it on the **glostylemcolalttree** style:

```
\setglossarystyle{mcolalttree}%
```

Put the navigation links in the header

```
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glssnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

`mcolalttreespannav` Similar to the `mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glssnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glssnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

3.8 Glossary Styles using supertabular environment (glossary-super.sty)

The glossary styles defined in the `glossary-super` package use the `supertabular` environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-super-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-super-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-super}[2022/10/14 v4.50 (NLCT)]
```

Requires the `supertabular` package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if `glossary-long` has been loaded.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if `glossary-long` has been loaded.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

super The super glossary style uses the `supertabular` environment (it uses lengths defined in the `glossary-long` package.)

```
\newglossarystyle{super}{%
```

Put the glossary in a `supertabular` environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail}%
  {\begin{supertabular}{lp{glsdescwidth}}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
```

```

\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}

```

superborder The **superborder** style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superborder}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|p{\glsdescwidth}|}%
{\end{supertabular}}%
}

```

superheader The **superheader** style is like the **super** style, but with a header:

```
\newglossarystyle{superheader}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns, a header and no tail:

```

\renewenvironment{theglossary}%
{\tablehead{\bfseries \entryname &
\bfseries\descriptionname\tabularnewline}%
\tabletail{}%
\begin{supertabular}{lp{\glsdescwidth}}%
{\end{supertabular}}%
}

```

superheaderborder The **superheaderborder** style is like the **super** style but with a header and border:

```
\newglossarystyle{superheaderborder}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns, a header and horizontal lines above and below the table:

```

\renewenvironment{theglossary}%
{\tablehead{\hline\bfseries \entryname &
\bfseries\descriptionname\tabularnewline\hline}%
\tabletail{\hline}
\begin{supertabular}{|l|p{\glsdescwidth}|}%
{\end{supertabular}}%
}

```

super3col The `super3col` style is like the `super` style, but with 3 columns:

```
\newglossarystyle{super3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{}}\tabletail{}}%  
  \begin{supertabular}{lp{\glstdescwidth}p{\glspagelistwidth}}}%  
  {\end{supertabular}}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%  
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} & ##2\tabularnewline  
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip  
  \renewcommand*{\glsgroupskip}{}%  
\else  
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%  
\fi  
}
```

super3colborder The `super3colborder` style is like the `super3col` style, but with a border:

```
\newglossarystyle{super3colborder}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline}\tabletail{\hline}}%
```

```

        \begin{supertabular}{|l|p{\glstdescwidth}|p{\glspagelistwidth|}}%
        {\end{supertabular}}}%
    }

```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```
\newglossarystyle{super3colheader}{%
```

Base it on the `glostylessuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```

\renewenvironment{theglossary}%
{
  \tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline}\tabletail{}}%
{
  \begin{supertabular}{|l|p{\glstdescwidth}|p{\glspagelistwidth|}}%
  {\end{supertabular}}}%
}

```

super3colheaderborder The `super3colheaderborder` style is like the `super3col` style but with a header and border:

```
\newglossarystyle{super3colheaderborder}{%
```

Base it on the `glostylessuper3colborder` style:

```
\setglossarystyle{super3colborder}%
```

Put the glossary in a `supertabular` environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
{
  \tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|p{\glstdescwidth}|p{\glspagelistwidth|}}%
  {\end{supertabular}}}%
}

```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```
\newglossarystyle{super4col}{%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```

\renewenvironment{theglossary}%
{
  \tablehead{}\tabletail{}%
  \begin{supertabular}{|l|l|l|l|}%
  \end{supertabular}}%

```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```


No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

super4colheader The `super4colheader` style is like the `super4col` but with a header row.

```
\newglossarystyle{super4colheader}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline}%
  \tabletail{}}%
  \begin{supertabular}{llll}%
  {\end{supertabular}}%
}
```

super4colborder The `super4colborder` style is like the `super4col` but with a border.

```
\newglossarystyle{super4colborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|1|1|1|1|}%
   {\end{supertabular}}%
}
```

super4colheaderborder The `super4colheaderborder` style is like the `super4col` but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the `glostylessuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
   \tabletail{\hline}%
   \begin{supertabular}{|1|1|1|1|}%
   {\end{supertabular}}%
}
```

altsuper4col The `altsuper4col` glossary style is like `super4col` but has provision for multiline descriptions.

```
\newglossarystyle{altsuper4col}{%
```

Base it on the `glostylessuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}%
   \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
   {\end{supertabular}}%
}
```

altsuper4colheader The `altsuper4colheader` style is like the `altsuper4col` but with a header row.

```
\newglossarystyle{altsuper4colheader}{%
```

Base it on the `glostylessuper4colheader` style:

```
\setglossarystyle{super4colheader}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\tabletail{}%
        \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
        {\end{supertabular}}%
    }

```

altsuper4colborder The `altsuper4colborder` style is like the `altsuper4col` but with a border.

```
\newglossarystyle{altsuper4colborder}{%
```

Base it on the `glostylesuper4colborder` style:

```
\setglossarystyle{super4colborder}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
    {\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}

```

altsuper4colheaderborder The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```
\newglossarystyle{altsuper4colheaderborder}{%
```

Base it on the `glostylesuper4colheaderborder` style:

```
\setglossarystyle{super4colheaderborder}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline
        \bfseries\entryname &
        \bfseries\descriptionname &
        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
    {\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}

```

3.9 Glossary Styles using supertabular environment (glossary-superragged.sty)

The glossary styles defined in the `glossary-superragged` package use the `supertabular` environment. These styles are like those provided by the `glossary-super` package, except that the multiline columns have ragged right justification.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-superragged-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-superragged-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2022/10/14 v4.50 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the supertabular package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`superragged` The superragged glossary style uses the supertabular environment.

```
\newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail}%
  {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
```

superraggedborder The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsgdescwidth}}}%
  {\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{\glsgdescwidth}}}%
  {\end{supertabular}}%
```

`superraggedheaderborder` The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```
\newglossarystyle{superraggedheaderborder}{%
```

Base it on the `glostylessuper` 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{\glstdescwidth}}}%
{\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{\glstdescwidth}%
>{\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]{%
\glstentryitem{##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{\glstdescwidth}|%
    >{\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{\glstdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)


```

\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & & \tabularnewline}%
\fi
}

```

altsuperragged4colheader The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{1>\raggedright}p{\glsgdescwidth}l%
>\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
}

```

altsuperragged4colborder The `altsuperragged4colborder` style is like the `altsuperragged4col` style but with a border.

```
\newglossarystyle{altsuperragged4colborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuper4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{
|l|>\raggedright}p{\glsgdescwidth}l|l|%
>\raggedright}p{\glspagelistwidth}l|l|}%
{\end{supertabular}}%
}

```

superragged4colheaderborder The `altsuperragged4colheaderborder` style is like the `altsuperragged4col` style but with a header and border.

```
\newglossarystyle{altsuperragged4colheaderborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}%
    {\l|>\raggedright}p{\glsdescwidth}\l|{%
      >\raggedright}p{\glspagelistwidth}\l|}%
  {\end{supertabular}}%
}
```

3.10 Tree Styles (glossary-tree.sty)

The `glossary-tree` style file defines glossary styles that have a tree-like structure. These are designed for hierarchical glossaries.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-tree-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-tree-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossary-tree}[2022/10/14 v4.50 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
```

`\glstreenamefmt` Format used to display the name in the tree styles. (This may be counteracted by `\glstnamefont`.) This command was previously also used to format the group headings.

```
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
```

`\glstreesubitem` Level 1 item used in index style.

```
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
```

`\glstreesubsubitem` Level 1 item used in index style.

```
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `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\glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
```

Sub entries: level 1 entries use `\subitem`, levels greater than 1 use `\subsubitem`. The level (`##1`) shouldn't be 0, as that's catered by `\glossentry`, but for completeness, if the level is 0, `\item` is used. The name is put in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \ifcase##1\relax
    % level 0
    \item
  \or
    % level 1
    \subitem
    \glssubentryitem{##2}%
  \else
    % all other levels
    \subsubitem
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\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{\glsgetgroupitle{##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*{\glsgrouphheading}[1]{}%
```

Main (level 0) entries: the name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstryitem{##1}\glstreenamfmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name and symbol are omitted. The description followed by the page list are displayed.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

treenonamegroup Like the `treenoname` style but the glossary groups have headings.

```
\newglossarystyle{treenonamegroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Give each group a heading:

```
\renewcommand{\glsgrouphheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsggetgrouptitle{##1}}\par\indexspace}%
}
```

treenonamehypergroup The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treenonamehypergroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreenameheaderfmt
{\glsnahypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

`\glsfindwidesttoplevelname` Find the widest name over all parentless entries in the given glossary or glossaries.

```
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forall glossaries[#1]{\@gls@type}%
{%
\forall glsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%%
}%%
\settowidth{\dimen@}%
{\glstreenamefmt{\glsentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@gls@widestname}{glo@\glsdetoklabel{\@glo@label}@name}%
\fi
}%%
}%%
}
```

`\glssetwidest` `\glssetwidest[level]{text}` sets the widest text for the given level. It is used by the `alttree` glossary styles to determine the indentation of each level.

```
\newcommand*{\glssetwidest}[2][0]{%
\expandafter\def\csname @gls@widestname\romannumeral#1\endcsname{%
#2}%
}
```

`\@gls@widestname` Initialise `\@gls@widestname`.

```
\newcommand*{\@gls@widestname}{}
```

`\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 `\@gls@widestname` which is set using `\glssetwidest`.

```
\newglossarystyle{alttree}{%
```


Redefine theglossary environment.

```
\renewenvironment{theglossary}%  
  {\def\@gls@prevlevel{-1}%  
   \mbox{}\par}%  
  {\par}%
```

Set the header and group headers to nothing.

```
\renewcommand*\glossaryheader{}%  
\renewcommand*\glsgroupheading[1]{}%
```

Redefine the way that the level 0 entries are displayed.

```
\renewcommand{\glossentry}[2]{%  
  \ifnum\@gls@prevlevel=0\relax  
  \else
```

Find out how big the indentation should be by measuring the widest entry.

```
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%  
  \fi
```

Set the hangindent and paragraph indent.

```
\hangindent\glstreeindent  
\parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
\makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%  
  \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
\def\@gls@prevlevel{0}%  
}%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
\ifnum##1=1\relax  
  \glssubentryitem{##2}%  
\fi
```

If the level hasn't changed, keep the same settings, otherwise adjust \glstreeindent accordingly.

```
\ifnum\@gls@prevlevel=##1\relax  
\else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
\ifundefined{@glswidestname\romannumeral##1}{%
  \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
  \settowidth{\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}{%
  \settowidth{\glstreeindent}{\glstreenamefmt{%
    \@glswidestname\space}}{%
  \settowidth{\glstreeindent}{\glstreenamefmt{%
    \csname @glswidestname\romannumeral\@gls@prevlevel
    \endcsname\space}}}%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
```

Set the hanging indentation.

```
\hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
\def\@gls@prevlevel{##1}%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

`alttreegroup` Like the `alttree` style but the glossary groups have headings.

```
\newglossarystyle{alttreegroup}{%  
Base it on the glosstylealttree style:  
  \setglossarystyle{alttree}%  
Give each group a heading.  
  \renewcommand{\glsgroupheading}[1]{\par  
    \def\@gls@prevlevel{-1}%  
    \hangindent0pt\relax  
    \parindent0pt\relax  
    \glstreegroupheaderfmt{\glsgrouptitle{##1}}%  
    \par\indexspace}%  
}
```

`alttreehypergroup` The `alttreehypergroup` style is like the `alttreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{alttreehypergroup}{%  
Base it on the glosstylealttree style:  
  \setglossarystyle{alttree}%  
Put the navigation links in the header  
  \renewcommand*{\glossaryheader}{%  
    \par  
    \def\@gls@prevlevel{-1}%  
    \hangindent0pt\relax  
    \parindent0pt\relax  
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%  
Put a hypertarget at the start of each group  
  \renewcommand*{\glsgroupheading}[1]{%  
    \par  
    \def\@gls@prevlevel{-1}%  
    \hangindent0pt\relax  
    \parindent0pt\relax  
    \glstreegroupheaderfmt  
      {\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par  
    \indexspace}}
```

4 Multi-Lingual Support

4.1 Babel Captions (`glossaries-babel.sty`)

Many thanks to everyone who contributed to the translations both via email and on `comp.text.tex`. Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-babel-2020-03-19.sty}  
\DeclareRelease{v4.49}{2021-11-01}{glossaries-babel-2021-11-01.sty}  
\DeclareCurrentRelease{v4.50}{2022-10-14}
```

Declare package

```
\ProvidesPackage{glossaries-babel}[2022/10/14 v4.50 (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{v4.50}{2022-10-14}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2022/10/14 v4.50 (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{v4.50}{2022-10-14}
```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2022/10/14 v4.50 (NLCT)]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

Load glossaries:

```
\RequirePackage{glossaries}
```

Add the new keys:

```
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
```

Add them to \@gls@keymap:

```
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}}%
}
```

Set the default values:

```
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
  \let\@glo@entryprefixfirstplural\@gls@default@value
}
```

Set the assignment code:

```
\appto\@newglossaryentryposthook{%
  \gls@assign@field{}\@glo@label}{prefix}\@glo@entryprefix}%
  \gls@assign@field{}\@glo@label}{prefixplural}\@glo@entryprefixplural}%
}
```

If prefixfirst has not been supplied, make it the same as prefix.

```
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefix\endcsname}\@glo@label}{prefixfirst}%
  {\@glo@entryprefixfirst}%
}
```

If prefixfirstplural has not been supplied, make it the same as prefixplural.

```
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @prefixplural\endcsname}\@glo@label}%
  {prefixfirstplural}\@glo@entryprefixfirstplural}%
}
```

Define commands to access these fields:

```
\glsentryprefixfirst
\newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}

\glsentryprefixfirstplural
\newcommand*{\glsentryprefixfirstplural}[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}

\glsentryprefix
\newcommand*{\glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}

\glsentryprefixplural
\newcommand*{\glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
```

Now for the initial upper case variants:

```
\Glsentryprefixfirst
\newcommand*{\Glsentryprefixfirst}[1]{%
  \@Glsentryfield{#1}{prefixfirst}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixfirst}{\Glentryprefixfirst}
```

\Glentryprefixfirstplural

```
\newcommand*{\Glentryprefixfirstplural}[1]{%  
  \@Glentryfield{#1}{prefixfirstplural}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixfirstplural}{\Glentryprefixfirstplural}
```

\Glentryprefix

```
\newcommand*{\Glentryprefix}[1]{%  
  \@Glentryfield{#1}{prefix}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefix}{\Glentryprefix}
```

\Glentryprefixplural

```
\newcommand*{\Glentryprefixplural}[1]{%  
  \@Glentryfield{#1}{prefixplural}%  
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glentryprefixplural}{\Glentryprefixplural}
```

Define commands to determine if the prefix keys have been set:

\ifglshasprefix

```
\newcommand*{\ifglshasprefix}[3]{%  
  \ifcempty{glo@\glsetoklabel{#1}@prefix}%  
    {#3}%  
    {#2}%  
}
```

\ifglshasprefixplural

```
\newcommand*{\ifglshasprefixplural}[3]{%  
  \ifcempty{glo@\glsetoklabel{#1}@prefixplural}%  
    {#3}%  
    {#2}%  
}
```

\ifglshasprefixfirst

```
\newcommand*{\ifglshasprefixfirst}[3]{%  
  \ifcempty{glo@\glsetoklabel{#1}@prefixfirst}%  
    {#3}%  
    {#2}%  
}
```

```

\ifglshasprefixfirstplural
    \newcommand*{\ifglshasprefixfirstplural}[3]{%
        \ifcempty{glo@glsdetoklabel{#1}@prefixfirstplural}%
        {#3}%
        {#2}%
    }

\@glsprefix@record@hook Need to take into account the possibility that glossaries-extra might be loaded
with the record option.
    \providecommand{\@glsprefix@record@hook}[2]{%
        \ifdef\@glsextr@record
        {\@glsextr@record{#1}{#2}{glslink}}%
        {}%
    }

\glsprefixsep Separator between prefix and term. Does nothing by default.
    \newcommand{\glsprefixsep}{}

    Define commands that insert the prefix before commands like \gls:

    \pgls
        \newrobustcmd{\pgls}{\@glshyp@opt\@pgls}

    \@pgls Unstarred version.
        \newcommand*{\@pgls}[2][{}]{%
            \new@ifnextchar[%
            {\@pgls@{#1}{#2}}%
            {\@pgls@{#1}{#2}[]}%
        }

    \@pgls@ Read in the final optional argument:
        \def\@pgls@#1#2[#3]{%
            \@glsprefix@record@hook{#1}{#2}%
            \glstoifexists{#2}%
            {%
                \ifglused{#2}%
                {%
                    \ifglshasprefix{#2}{\glstryprefix{#2}\glsprefixsep}{}%
                }%
                {%
                    \ifglshasprefixfirst{#2}{\glstryprefixfirst{#2}\glsprefixsep}{}%
                }%
                \@glso{#1}{#2}[#3]%
            }%
        }

```

Similarly for the plural version:


```
\pglsp1
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
```

\@pglsp1 Unstarred version.

```
\newcommand*{\@pglsp1}[2][{}]{%
\new@ifnextchar[%
{\@pglsp1@{#1}{#2}}%
{\@pglsp1@{#1}{#2}[]}%
}
```

\@pglsp1@ Read in the final optional argument:

```
\def\@pglsp1@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifglshasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
}%
{%
\ifglshasprefixfirstplural{#2}%
{\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
}%
\@glspl@{#1}{#2}[#3]%
}%
}
```

Now for the first letter upper case versions:

```
\Pgls
\newrobustcmd{\Pgls}{\@gls@hyp@opt\@Pgls}
```

Add sentence case mapping.

```
\glsmfuaddmap{\pgls}{\Pgls}
```

\@Pgls Unstarred version.

```
\newcommand*{\@Pgls}[2][{}]{%
\new@ifnextchar[%
{\@Pgls@{#1}{#2}}%
{\@Pgls@{#1}{#2}[]}%
}
```

\@Pgls@ Read in the final optional argument:

```
\def\@Pgls@#1#2[#3]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
}
```

```

{%
  \ifglshasprefix{#2}%
  {%
    \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}%
    {%
      \ifgls hasprefixplural{#2}%
      {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%
      \ifgls hasprefixfirstplural{#2}%
      {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{%
    }%
    \@GSLSp1@{#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{v4.50}{2022-10-14}
```

Package version number now in line with main glossaries package number.

```
\ProvidesPackage{glossaries-accsupp}[2022/10/14 v4.50 (NLCT)
  Experimental glossaries accessibility]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```
\@ifpackageloaded{glossaries-extra}
{%
```

If the `accsupp` option was used, `\@glxtr@doaccsupp` will have been set, otherwise it will be empty.

```
\ifx\@glxtr@doaccsupp\empty
\GlossariesWarning{The ‘glossaries-accsupp’
package has been loaded\MessageBreak
after the ‘glossaries-extra’ package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the ‘accsupp’ option when you load\MessageBreak
‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
before loading ‘glossaries-extra’}%
\fi
}
{}
```

`\compatibleglossentry` Override style compatibility macros:

`\compatiblesubglossentry`

Required packages:

```
\RequirePackage{glossaries}
```

`\gls@accsupp@engine` There’s currently only support for `accsupp`, but if you define `\gls@accsupp@engine` before loading `glossaries-accsupp`, you can prevent `accsupp` from being loaded. Redefining this command after `glossaries-accsupp` has loaded obviously won’t do anything (and so is an internal command to deter casual use). If it is defined to something other than `accsupp` then `\gls@accessibility` will need to be defined to something appropriate.

```
\providecommand{\gls@accsupp@engine}{accsupp}
```

```
\gls@accessibility[\options]{\PDF element}{\value}{\content}
```

`\gls@accessibility`

```
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
```

```
\glsaccessibility[\options]{\PDF element}{\value}{\content}
```

`\glsaccessibility`

User-level command that includes debug info if required.

```
\newcommand{\glsaccessibility}[4][]{%  
  \@glsshowacccsupp{#1}{#2}{#3}%  
  \gls@accessibility{#1}{#2}{#3}{#4}%  
}
```

6.1 Defining Replacement Text

The version 0.1 stored the replacement text in the symbol key. This has been changed to use the new keys defined here. Example of use:

```
%\newglossaryentry{dr}{name=Dr,description={},access={Doctor}}  
%
```

access The replacement text corresponding to the name key:

```
\define@key{glossentry}{access}{%  
  \def\@glo@access{#1}%  
}
```

textaccess The replacement text corresponding to the text key:

```
\define@key{glossentry}{textaccess}{%  
  \def\@glo@textaccess{#1}%  
}
```

firstaccess The replacement text corresponding to the first key:

```
\define@key{glossentry}{firstaccess}{%  
  \def\@glo@firstaccess{#1}%  
}
```

pluralaccess The replacement text corresponding to the plural key:

```
\define@key{glossentry}{pluralaccess}{%  
  \def\@glo@pluralaccess{#1}%  
}
```

firstpluralaccess The replacement text corresponding to the firstplural key:

```
\define@key{glossentry}{firstpluralaccess}{%  
  \def\@glo@firstpluralaccess{#1}%  
}
```

symbolaccess The replacement text corresponding to the symbol key:

```
\define@key{glossentry}{symbolaccess}{%  
  \def\@glo@symbolaccess{#1}%  
}
```

symbolpluralaccess The replacement text corresponding to the symbolplural key:

```
\define@key{glossentry}{symbolpluralaccess}{%  
  \def\@glo@symbolpluralaccess{#1}%  
}
```

descriptionaccess The replacement text corresponding to the description key:

```
\define@key{glossentry}{descriptionaccess}{%  
  \def\@glo@descaccess{#1}%  
}
```

descriptionpluralaccess The replacement text corresponding to the descriptionplural key:

```
\define@key{glossentry}{descriptionpluralaccess}{%  
  \def\@glo@descpluralaccess{#1}%  
}
```

shortaccess The replacement text corresponding to the short key:

```
\define@key{glossentry}{shortaccess}{%  
  \def\@glo@shortaccess{#1}%  
}
```

shortpluralaccess The replacement text corresponding to the shortplural key:

```
\define@key{glossentry}{shortpluralaccess}{%  
  \def\@glo@shortpluralaccess{#1}%  
}
```

longaccess The replacement text corresponding to the long key:

```
\define@key{glossentry}{longaccess}{%  
  \def\@glo@longaccess{#1}%  
}
```

longpluralaccess The replacement text corresponding to the longplural key:

```
\define@key{glossentry}{longpluralaccess}{%  
  \def\@glo@longpluralaccess{#1}%  
}
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%  
  \def\@glo@useriaccess{#1}%  
}
```

user2access The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%  
  \def\@glo@useriiaccess{#1}%  
}
```

user3access The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%  
  \def\@glo@useriiiaccess{#1}%  
}
```

`user4access` The replacement text corresponding to the `user4` key:

```
\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
```

`user5access` The replacement text corresponding to the `user5` key:

```
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the `user6` key:

```
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%

  {descriptionaccess}{descaccess},%
  {descriptionpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the access key was initialised to the value of the `symbol` key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for `symbol` but not for `name` so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
```



```

\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
\def\@glo@access{\relax}%

```

Initialise the other keys:

```

\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
\def\@glo@longpluralaccess{\@glo@longaccess}%
\def\@glo@useriaccess{\relax}%
\def\@glo@useriiaaccess{\relax}%
\def\@glo@useriiiaaccess{\relax}%
\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}

```

Add to the end hook:

```

\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook

```

Store the access information:

```

\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
  \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
  \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
  \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
  \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
  \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
  \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%

```

```

        \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaaccess\endcsname{%
    \@glo@useriiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaaccess\endcsname{%
    \@glo@useriiiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
    \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
    \@glo@userviaccess}%
}

```

6.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

\newcommand*{\glsentryaccess}[1]{%
    \@gls@entry@field{#1}{access}%
}

```

`\glsentrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

\newcommand*{\glsentrytextaccess}[1]{%
    \@gls@entry@field{#1}{textaccess}%
}

```

}

`\glsentryfirstaccess` Get the value of the `firstaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
```

`\glsentrypluralaccess` Get the value of the `pluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
```

`\glsentryfirstpluralaccess` Get the value of the `firstpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
```

`\glsentrysymbolaccess` Get the value of the `symbolaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
```

`\glsentrysymbolpluralaccess` Get the value of the `symbolpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
```

`\glsentrydescaccess` Get the value of the `descriptionaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
```

`\glsentrydescpluralaccess` Get the value of the `descriptionpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
```

`\glsentryshortaccess` Get the value of the `shortaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
```

`\glsentryshortpluralaccess` Get the value of the `shortpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
```

`\glentrylongaccess` Get the value of the `longaccess` key for the entry with the given label:

```

\newcommand*{\glentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

`\glentrylongpluralaccess` Get the value of the `longpluralaccess` key for the entry with the given label:

```

\newcommand*{\glentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}

```

`\glentryuseriaccess` Get the value of the `user1access` key for the entry with the given label:

```

\newcommand*{\glentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}

```

`\glentryuseriiaccess` Get the value of the `user2access` key for the entry with the given label:

```

\newcommand*{\glentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}

```

`\glentryuseriiiaccess` Get the value of the `user3access` key for the entry with the given label:

```

\newcommand*{\glentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}

```

`\glentryuserivaccess` Get the value of the `user4access` key for the entry with the given label:

```

\newcommand*{\glentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}

```

`\glentryuservaccess` Get the value of the `user5access` key for the entry with the given label:

```

\newcommand*{\glentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}

```

`\glentryuserviaccess` Get the value of the `user6access` key for the entry with the given label:

```

\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}

```

There are three types of replacement text:

Alt Description of some content that's non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require `E`, the `symbol` key would require `ActualText`, and a field that contains an image would require `Alt`.

```
\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\glsfieldaccsupp`

Test if there’s a command called `\gls<field>accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If `glossaries-extra` has been loaded, this first checks for `\glsxtr<category><field>accsupp` and `\glsxtr<category>accsupp`.

```
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
```

```
\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\xglsfieldaccsupp`

As `\glsfieldaccsupp` but fully expand replacement text.

```
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
```

`\glsshortaccsupp`

```
\glsshortaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp`

```
\glsshortplaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

`\glsaccsupp`

```
\glsaccsupp{<replacement text>}{<text>}
```

```
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
```

`\xglsaccsupp` Fully expands replacement text before calling `\glsaccsupp`

```
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
```

`\@gls@access@display` Deprecated in v4.45. Use `\@gls@fieldaccess@display` instead.

```
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
```

`\@gls@fieldaccess@display`

```
\@gls@fieldaccess@display{<label>}{<field>}{<content>}
{<replacement>}
```

```
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
```

`\glsnameaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@access}%
```

```

        {#1}%
        {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
    }

\glstextaccessdisplay As above but for the textaccess replacement text.
\newrobustcmd*{\glstextaccessdisplay}[2]{%
    \ifcsundef{glo@glsglsetoklabel{#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@glsglsetoklabel{#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@glsglsetoklabel{#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@glsglsetoklabel{#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@glsglsetoklabel{#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@glsglsetoklabel{#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@glsglsetoklabel{#2}@descaccess}%

```

```

        {#1}%
        {\@gls@fieldaccess@display{#2}{desc}{#1}{\glentrydescaccess{#2}}}%
    }

\descriptionpluralaccessdisplay As above but for the descriptionpluralaccess replacement text.
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@descpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glentrydescpluralaccess{#2}}}%
}

\glsshortaccessdisplay As above but for the shortaccess replacement text.
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{short}{#1}{\glentryshortaccess{#2}}}%
}

\glsshortpluralaccessdisplay As above but for the shortpluralaccess replacement text.
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glentryshortpluralaccess{#2}}}%
}

\glslongaccessdisplay As above but for the longaccess replacement text.
\newrobustcmd*{\glslongaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{long}{#1}{\glentrylongaccess{#2}}}%
}

\glslongpluralaccessdisplay As above but for the longpluralaccess replacement text.
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glentrylongpluralaccess{#2}}}%
}

\glsuseriaccessdisplay As above but for the user1access replacement text.
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{useri}{#1}{\glentryuseriaccess{#2}}}%
}

\glsuseriiaccessdisplay As above but for the user2access replacement text.
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
    \ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%

```



```

    {#1}%
    {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}

```

`\glsuseriiiaccessdisplay` As above but for the `user3access` replacement text.

```

\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}

```

`\glsuserivaccessdisplay` As above but for the `user4access` replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
}

```

`\glsuservaccessdisplay` As above but for the `user5access` replacement text.

```

\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}

```

`\glsuserviaccessdisplay` As above but for the `user6access` replacement text.

```

\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@glstoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}

```

`\glsaccessdisplay` Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{}%
  }%
  {%
    \csname gls#1accessdisplay\endcsname{#2}{#3}%
  }%
}

```

`\@gls@default@entryfmt` Deprecated `\@gls@default@entryfmt` was redefined to use accessibility information but has been removed in v4.50.

```

\glsgenentryfmt Redefine to use accessibility information.
\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      Plural form
      \glscapscase
      {%
        Don't adjust case
        \ifglsused\glslabel
        {%
          Subsequent use
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
          }%
          {%
            First use
            \glsfirstpluralaccessdisplay
              {\glsentryfirstplural{\glslabel}}{\glslabel}%
            \glsinsert
            }%
            }%
            {%
              Make first letter upper case
              \ifglsused\glslabel
              {%
                Subsequent use.
                \glspluralaccessdisplay
                  {\Glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
                }%
                {%
                  First use
                  \glsfirstpluralaccessdisplay
                    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
                  \glsinsert
                  }%
                  }%
                  {%
                    Make all upper case
                    \ifglsused\glslabel
                    {%

```

Subsequent use

```
\glspluralaccessdisplay
  {\glsuppercase{\glsentryplural{\glslabel}}}%
  {\glslabel}%
\glsuppercase{\glsinsert}%
}%
{%
```

First use

```
\glsfirstpluralaccessdisplay
  {\glsuppercase{\glsentryfirstplural{\glslabel}}}%
  {\glslabel}%
\glsuppercase{\glsinsert}%
}%
}%
}%
{%
```

Singular form

```
\glscapscase
{%
```

Don't adjust case

```
\ifglused\glslabel
{%
```

Subsequent use

```
\glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
```

First use

```
\glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
}%
{%
```

Make first letter upper case

```
\ifglused\glslabel
{%
```

Subsequent use

```
\glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
```

First use

```
\glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
```

```

    }%
    {%
Make all upper case
    \ifglused\glslabel
    {%
Subsequent use
    \glstextaccessdisplay
    {\glsupercase{\glentrytext{\glslabel}}}{\glslabel}%
    \glsupercase{\glsinsert}%
    }%
    {%
First use
    \glsfirstaccessdisplay
    {\glsupercase{\glentryfirst{\glslabel}}}{\glslabel}%
    \glsupercase{\glsinsert}%
    }%
    }%
    }%
    {%
Custom text provided in \glsdisp. (The insert should be empty at this point.)
The accessibility information, if required, will have to be explicitly included in
the custom text.
    \glscustomtext\glsinsert
    }%
}

\glsgenacfmt Redefine to include accessibility information.
\renewcommand*{\glsgenacfmt}{%
    \ifdefempty\glscustomtext
    {%
        \ifglused\glslabel
        {%
Subsequent use:
            \glsifplural
            {%
Subsequent plural form:
                \glscapscase
                {%
Subsequent plural form, don't adjust case:
                    \acronymfont
                    {\glsshortpluralaccessdisplay
                    {\glentryshortpl{\glslabel}}{\glslabel}}%
                    \glsinsert
                    }%
                    {%

```

Subsequent plural form, make first letter upper case:

```
\acronymfont
{\glsshortpluralaccessdisplay
{\Glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent plural form, all caps:

```
\glssupercase
{\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:

```
\glssupercase
{\acronymfont{%
\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glscapscase
{%
```

First use plural form, don't adjust case:

```
\genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, make first letter upper case:

```
\Genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, all caps:

```
\glsuppercase
{\genplacrformat{\glslabel}{\glsinsert}}%
}%
}%
{%
```

First use singular form

```
\glscapscase
{%
```

First use singular form, don't adjust case:

```
\genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, make first letter upper case:

```
\Genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, all caps:

```
\glsuppercase
{\genacrformat{\glslabel}{\glsinsert}}%
}%
}%
}%
}%
{%
```

User supplied text. (The insert should be empty at this point.) The accessibility information, if required, will have to be explicitly included in the custom text.

```
\glscustomtext
}%
}
```

`\genacrformat` Redefine to include accessibility information.

```
\renewcommand*\genacrformat[2]{%
\glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
(\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
```

```

\Genacrfullformat Redefine to include accessibility information.
    \renewcommand*{\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
        (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
    }

\genplacrfullformat Redefine to include accessibility information.
    \renewcommand*{\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
    }

\Genplacrfullformat Redefine to include accessibility information.
    \renewcommand*{\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
    }

\@acrshort
    \def\@acrshort#1#2[#3]{%
        \glsdoifexists{#2}%
        {%
            \let\do@gl@link@checkfirsthyper\relax
            \let\glsifplural\@secondoftwo
            \let\glscapscase\@firstofthree
            \let\glsinsert\@empty
            \def\glscustomtext{%
                \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
            }%
        }

    Call \@gl@link
        \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
}

\@Acrshort
    \def\@Acrshort#1#2[#3]{%
        \glsdoifexists{#2}%
        {%
            \let\do@gl@link@checkfirsthyper\relax
            \let\glsifplural\@secondoftwo
            \let\glscapscase\@secondofthree
            \let\glsinsert\@empty
            \def\glscustomtext{%
                \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
            }%
        }
    }

```

```

Call \@gls@link
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@ACRshort
\def\@ACRshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\relax

        \let\glsifplural\@secondoftwo
        \let\glscapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsshortaccessdisplay
                {\glsuppercase{\glstryshort{#2}}}{#2}}#3%
        }%
    }%

Call \@gls@link
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@acrlong
\def\@acrlong#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\relax

        \let\glsifplural\@secondoftwo
        \let\glsapsase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glslongaccessdisplay{\glstrylong{#2}}{#2}}#3%
        }%
    }%

Call \@gls@link
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@Acrlong
\def\@Acrlong#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\relax

```



```

\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
}%

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@ACRlong
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \glsuppercase{\glsentrylong{#2}}}{#2}}#3}%
    }%

    Call \@gls@link
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
      }%

      \glspostlinkhook
    }

\@glstext@
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glsentrytext{#2}}{#2}}#3}%
}

\@Glstext@
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}}#3}%
}

\@GLStext@
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glstextaccessdisplay{\glsuppercase{\glsentrytext{#2}}}{#2}%
  \glsuppercase{#3}}%
}

```

```

\@glsfirst@
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}

\@Glsfirst@
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}

\@GLSfirst@
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\glsupercase{\glsentryfirst{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\glsupercase{\glsentryplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsfirstplural@
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}

\@Glsfirstplural@
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

\@GLSfirstplural@
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\glsupercase{\glsentryfirstplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsname@
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}

\@Glsname@
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}

\@GLSname@
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsnameaccessdisplay{\glsuppercase{\glsentryname{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdesc@
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}

\@Glsdesc@
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

\@GLSdesc@
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionaccessdisplay{\glsuppercase{\glsentrydesc{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdescplural@
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}

\@Glsdescplural@
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}

\@GLSdescplural@
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionpluralaccessdisplay{\glsuppercase{\glsentrydescplural{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

```

\@glssymbol@
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSymbol@
\def\@GLSymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolaccessdisplay{\glsupercase{\glentrysymbol{#2}}}{#2}%
    \glsupercase{#3}}}%
}

\@glssymbolplural@
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glentrysymbolplural{#2}}{#2}#3}%
}

\@Glsymbolplural@
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}

\@GLSymbolplural@
\def\@GLSymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolpluralaccessdisplay{\glsupercase{\glentrysymbolplural{#2}}}{#2}%
    \glsupercase{#3}}}%
}

\@glsuseri@
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glentryuseri{#2}}{#2}#3}%
}

\@Glsuseri@
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}

\@GLSuseri@
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\glsupercase{\glentryuseri{#2}}}{#2}%
    \glsupercase{#3}}}%
}

```

```

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}

\@GLSuserii@
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\glsupercase{\glsentryuserii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriii@
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}

\@Glsuseriii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}

\@GLSuseriii@
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\glsupercase{\glsentryuseriii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriv@
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}

\@Glsuseriv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}

\@GLSuseriv@
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\glsupercase{\glsentryuseriv{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsuserv@
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}

\@Glsuserv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

\@GLSuserv@
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuservaccessdisplay{\glsupercase{\glsentryuserv{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuservi@
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}

\@Glsuservi@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}

\@GLSuservi@
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserviaccessdisplay{\glsupercase{\glsentryuservi{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

```

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

```

\accsuppglossaryentryfield

```

\newcommand*{\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}

```

\accsuppglossarysubentryfield

```

\newcommand*{\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}

```

6.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short $\langle long \rangle$ ($\langle short \rangle$) acronym style.

```
\renewacronymstyle{long-short}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}{##1})}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1})}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glseyentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```

short-long $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\renewacronymstyle{short-long}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1})}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1})}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1})}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glseyentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```



```

\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}{\glpluralsuffix}%
}

```

`long-short-desc` $\langle long \rangle$ ($\langle short \rangle$) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
  }

```

`long-sc-short-desc` $\langle long \rangle$ ($\textsc{\langle short \rangle}$) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%

```

```

{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
\renewacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

short-long-desc <short> ({<long>}) acronym style that has an accompanying description (which
the user needs to supply).
\renewacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

sc-short-long-desc <long> (\textsc{<short>}) acronym style that has an accompanying description
(which the user needs to supply).
\renewacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}

```

sm-short-long-desc *<long>* (`\textsmaller{<short>}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sm-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}

```

dua *<long>* only acronym style.

```

\renewacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
\ifglshaslong{\glslabel}%
{%
\glsifplural
{%

```

Plural form:

```

\glscapscase
{%

```

Plural form, don't adjust case:

```

\glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
\glsinsert
}%
{%

```

Plural form, make first letter upper case:

```

\glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
\glsinsert
}%
{%

```

Plural form, all caps:

```

\glslongpluralaccessdisplay
{\glsupercase{\glsentrylongpl{\glslabel}}}{\glslabel}%

```

```

        \glssupercase{\glsinsert}%
    }%
}%
{%
Singular form
        \glscapscase
    {%
Singular form, don't adjust case:
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, make first letter upper case:
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, all caps:
        \glslongaccessdisplay
        {\glssupercase
         {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
        \glssupercase{\glsinsert}%
    }%
}%
{%
Not an acronym:
        \glsgenentryfmt
    }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glslongaccessdisplay
        {\glssupercase{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%

```

```

\glslongpluralaccessdisplay
  {\glsentrylongpl{##2}}{##2}##3\space
(\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsupercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%

```

}%

footnote *<short>*\footnote{*<long>*} acronym style.

\renewacronymstyle{footnote}%

{%

Check for long form in case this is a mixed glossary.

\ifglshaslong{\glslabel}{\glsacronym}{\glsentryfmt}%

}%

{%

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

Need to ensure hyperlinks are switched off on first use:

\glshyperfirstfalse

\renewcommand*{\genacrfullformat}[2]{%

\glsshortaccessdisplay

{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%

\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%

}%

\renewcommand*{\Genacrfullformat}[2]{%

\glsshortaccessdisplay

{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%

\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%

}%

\renewcommand*{\genplacrfullformat}[2]{%

\glsshortpluralaccessdisplay

{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%

\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%

}%

\renewcommand*{\Genplacrfullformat}[2]{%

\glsshortpluralaccessdisplay

{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%

\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%

}%

\renewcommand*{\acronymentry}[1]{%

\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%

\renewcommand*{\acronymsort}[2]{##1}%

\renewcommand*{\acronymfont}[1]{##1}%

\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%

Don't use footnotes for \acrfull:

\renewcommand*{\acrfullfmt}[3]{%

\glslink[##1]{##2}{%

\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space

(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%

\renewcommand*{\Acrfullfmt}[3]{%

\glslink[##1]{##2}{%

\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space

(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%

\renewcommand*{\ACRfullfmt}[3]{%

\glslink[##1]{##2}{%

```

\glsshortaccessdisplay
  {\glssupercase
    {\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\glssupercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}}}}}%

```

Similarly for `\glentryfull` etc:

```

\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1}}}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1}}}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}}}%
}

```

`footnote-sc \textsc{<short>}\footnote{<long>}` acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
```



```
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}
```

```
\glsglsdefaultshortaccess{<long>}{<short>}
```

`\glsglsdefaultshortaccess`

Default shortaccess value.

```
\newcommand*{\glsglsdefaultshortaccess}[2]{#1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*{\newacronymhook}{%
\protected@edef\@gls@keylist{%
shortaccess={\glsglsdefaultshortaccess{the\glslongtok}{the\glsshorttok}},%
shortpluralaccess={\glsglsdefaultshortaccess{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` `xindy` and `makeindex` formatting, so can be used with old documents that had customized style files, but hyperlinks may not work properly.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-207-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-207-2021-11-01.sty}
\DeclareCurrentRelease{v4.50}{2022-10-14}

```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2022/10/14 v4.50 (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{v4.50}{2022-10-14}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2022/10/14 v4.50 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}  
{glossaries-compatible-307.sty no longer available. Use rollback if you need it}%  
{Try \string\usepackage{glossaries}[=v4.46]}
```

8 Rollback

8.1 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]  
\RequirePackage{ifthen}  
\RequirePackage{xkeyval}[2006/11/18]  
\RequirePackage{mfirstuc}  
\RequirePackage{textcase}  
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%  
\RequirePackage{xfor}  
\RequirePackage{datatool-base}  
\RequirePackage{amsgen}  
\RequirePackage{etoolbox}  
\newif\if@gls@docloaded  
\@ifpackageloaded{doc}%  
{%  
  \@gls@docloadedtrue  
}%  
{%  
  \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%  
}  
\if@gls@docloaded  
  \let\glsorg@theglossary\theglossary  
  \let\glsorg@endtheglossary\endtheglossary  
  \let\glsorg@PrintChanges\PrintChanges  
  \renewcommand{\PrintChanges}{%  
    \begingroup  
      \let\theglossary\glsorg@theglossary  
      \let\endtheglossary\glsorg@endtheglossary  
      \glsorg@PrintChanges  
    \endgroup  
  }  
\fi
```

```

\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\gls@debug@nr\relax
% debug=true
\@gls@debugtrue
\renewcommand*{\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*{\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glsshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*{\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*{\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*{\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*{\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*{\glsshowtarget}[1]{%
\ifmmode
\nfss@text{\glsshowtargetfont [#1]}%
\else
\ifinner
{\glsshowtargetfont [#1]}%

```

```

\else
\glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
\glsshowtargetsymbolsmarginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbols{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*{\@glsshowtarget}[1]{%
\begingroup
\protected@edef\@gls@tmp{#1}%
\@onelevel@sanitize\@gls@tmp
\expandafter\glsshowtarget\expandafter{\@gls@tmp}%
\endgroup
}
\newcommand*\@glsshowaccsupp}[3]{%
\newrobustcmd*{\glsshowaccsupp}[3]{%
\ifstrempy{#1}%
{\@glsshowtarget{/#2 (#3)}}%
{\@glsshowtarget{/#2 (#3) [#1]}}%
}
}
\newcommand*\@gls@see@noindex{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries\space (or move
\string\newglossaryentry\space
definitions into the preamble)}}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key. It may
be that the 'see' key has been written to the .glsdefs
file from the previous run, in which case you need to
move your definitions
to the preamble if you don't want to use
\string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinindex}%
[\gls@seenoinindex@val\gls@seenoinindex@nr]{error,warn,ignore}{%
\ifcase\gls@seenoinindex@nr
\renewcommand*\@gls@see@noindex{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key}%
}%
}

```

```

\or
\renewcommand*{\@gls@see@noindex}{%
  \GlossariesWarning{'\gls@xr@key' key ignored}%
}%
\or
\renewcommand*{\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{}%
\renewcommand*{\@glossarysec}{#1}%
\newcommand*{\@glossarysecstar}{*}%
\newcommand*{\@glossaryseclabel}{}%
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{}%
false,nolabel,autolabel,nameref}[nolabel]{}%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{}%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```

```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*{\@glo@seeautonumberlist}{%
    \def\@glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\gls@esclocationstrue
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}%
\IfFileExists{supertabular.sty}{%
  \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}[=v4.46]}%
  \newcommand*\@gls@loadsuper{}}%
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}%
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}%
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
    \ifdefstring{\@glossary@default@style}{list}%
    {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}%
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*\@gls@postdescription{%
  \ifgls@postdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\gls@nopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\gls@nogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}{%
  \gls@ucmarktrue
}%
{%
  \gls@ucmarkfalse
}

```



```

}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
    \ifundef\c@glossaryentry
      {%
        \ifx\@gls@counterwithin\@empty
          \newcounter{glossaryentry}%
        \else
          \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
      }%
    }%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
    {%
      \ifglssubentrycounter
        \ifglentrycounter
          \newcounter{glossarysubentry}[glossaryentry]%
        \else
          \newcounter{glossarysubentry}%
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
      \fi
    }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
  \fi
  \let\glsdosanitizesort\@gls@sanitizesort
  \glsprestandardsort{\@glo@sort}{##1}{##2}%
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
\newcommand*\@gls@setupsort@def{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \ifcsundef{glossary@##1@sortcount}%
  {\@gls@defsortcount{##1}}%
  {}%
  \expandafter\global\expandafter
  \advance\csname glossary@##1@sortcount\endcsname by 1\relax
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
    \expandafter\glsortnumberfmt
    {\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%

```

```

\edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\edef\@glo@type{\csname glo@##1@type\endcsname}%
\edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected\xdef\csname glo@##1@sort\endcsname{%
\expandafter\glsortnumberfmt
{\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[gls2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}{gls}{acronym}[true]{%
\ifglsacronym
\renewcommand{\@gls@do@acronymsdef}{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%

```

```

\renewcommand*{\acronymtype}{acronym}%
\newcommand*{\gls@tr@set@acronym@toctitle}{%
  \translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
\else
\let\@gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
\ifglsacronym
\ifbool{glscompatible-3.07}%
{%
{%
\providecommand*{\printacronyms}[1][{}]{%
\printglossary[type=\acronymtype,#1]}%
}%
\fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
\glsacronymtrue
\def\@gls@do@acronymsdef{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*{\acronymtype}{acronym}%
\newcommand*{\gls@tr@set@acronym@toctitle}{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
}
\newcommand*{\@glsacronymlists}{}
\newcommand*{\@addtoacronymlists}[1]{%
\ifx\@glsacronymlists\@empty
\protected@xdef\@glsacronymlists{#1}%
\else
\protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
\fi
}
\newcommand*{\DeclareAcronymList}[1]{%
\glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
\edef\@do@gls@islistofacronyms{%
\noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
\@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
\def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
\def\@gls@before{##1}\def\@gls@after{##2}}%

```

```

\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
  #4%
\else
  #3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@gls@acronymlists}{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
  \fi
}

```

```

\renewcommand*\GlossariesWarningNoLine}[1]{}%
\renewcommand*\glosortentrieswarning{}%
\renewcommand*\@gls@missinglang@warn}[2]{}%
\fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘#1’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-#2’ or similar}%
}
\@gls@declareoption{nolangwarn}{%
\renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined{%
\GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheGLOSSdefined{%
\GlossariesWarning{Overriding ‘theGLOSS’ environment}%
}
\@gls@declareoption{noredefwarn}{%
\renewcommand*\@gls@warnonglossdefined{}%
\renewcommand*\@gls@warnontheGLOSSdefined{}%
}
\newcommand*\@gls@sanitizedesc{%
}
\newcommand*\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename{}
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort{%
\ifgls@sanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort

```

```

\fi
}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefined\@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}{%
  \ifdefined\@glo@sort
  {%
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\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{\language}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\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\@glsl@labelsfile
    \immediate\openout\@glsl@labelsfile=\jobname.glslabels
    \forallglslentries[\@gls@types,\@ignored@glossaries]{\@glslentry}%
    {\ifdefempty{\@glslentry}{\immediate\write\@glsl@labelsfile{\@glslentry}}}%
    \immediate\closeout\@glsl@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglslautomake
\newcommand{\glsl@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
    [\glsl@automake@val\glsl@automake@nr]{true,false,immediate}[true]{%
    \ifnum\glsl@automake@nr=1\relax
        \glslautomakefalse
    \else
        \glslautomaketrue
    \fi

```

```

\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \PackageError{glossaries}{You must use
      \string\makeglossaries\space with automake=true}
    {%
      Either remove the automake=true setting or
      add \string\makeglossaries\space to your document preamble.%
    }%
  }%
\else
  \renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifglssavewrites
    \renewcommand*{\@gls@writefiles}{\@gls@writefiles}%
  \else
    \let\@gls@writefiles\@empty
  \fi
}
\glssavewritesfalse
\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][\@gls@tr@set@symbols@toctitle]{%
\printglossary[type=symbols,##1]}%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef

```

```

}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\]{\printglossary[type=index,##1]}%
  \newcommand*{\newterm}[2][\]{%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \expandtwoargs
    \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@option
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.46]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts

```



```

        \DefineAcronymSynonyms
    \fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifgl Sentrycounter
    \let\@gls@doentrycounterdef\relax
\else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifgl subentrycounter
    \let\@gls@dosubentrycounterdef\relax
\else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
    \ifcsundef{chapter}{}%
    {%
        \let\@gls@old@chapter\@chapter
        \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
        \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}%
    }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{}
\newcommand*{\@onlypremakeg}[1]{%
    \ifx\@gls@onlypremakeg\@empty
        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{}
\@for\@thiscs:=\@gls@onlypremakeg\do{%

```

```

\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glsettoctitle}[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*{\glnumbersgroupname}{Numbers}
\newcommand*{\glspluralsuffix}{s}
\newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*{\glsupacrpluralsuffix}{\glsup{t}\glsacrpluralsuffix}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
\newcommand*{\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*{\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp

```

```

        \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
}%
}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
  }-}%
\fi
\DeclareRobustCommand*{\nopostdesc}{%
\newcommand*{\@nopostdesc}{%

```

```

\let\org@glspostdescription\glspostdescription
\def\glspostdescription{%
  \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gl@istfilebase}{#1}%
  \ifglxsindy
    \def\istfilename{\gl@istfilebase.xdy}
  \else
    \def\istfilename{\gl@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxsindy
  \def\istfilename{\gl@istfilebase.xdy}
\else
  \def\istfilename{\gl@istfilebase.ist}
\fi
\newcommand*{\gl@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{%
\newcommand*{\glscpositor}{.}
\newcommand*{\glSetCompositor}[1]{%
  \renewcommand*{\glscpositor}{#1}}
\@onlypremakeg\glSetCompositor
\newcommand*{\@gl@Alphacompositor}{\glscpositor}
\ifglxsindy
  \newcommand*{\glSetAlphaCompositor}[1]{%
    \renewcommand*{\@gl@Alphacompositor}{#1}}
\else
  \newcommand*{\glSetAlphaCompositor}[1]{%
    \gl@noxindywarning\glSetAlphaCompositor}
\fi
\@onlypremakeg\glSetAlphaCompositor
\newcommand*{\gl@suffixF}{%
\newcommand*{\glSetSuffixF}[1]{%
  \renewcommand*{\gl@suffixF}{#1}}
\@onlypremakeg\glSetSuffixF
\newcommand*{\gl@suffixFF}{%
\newcommand*{\glSetSuffixFF}[1]{%
  \renewcommand*{\gl@suffixFF}{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\gl@numberformat}[1]{#1}%
}%
{%
  \newcommand*{\gl@numberformat}[1]{\glshypernumber{#1}}%
}

```

```

\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%

```

```

\else
  \@mkboth{#1}{#1}%
\fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*\@glossarysection[2]{%
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname[#1]{#2}%
  }%
  {%
    \csname\@glossarysec\endcsname*{#2}%
    \gls@toc{#1}{\@glossarysec}%
  }%
  \@glossaryseclabel
}
\newcommand*\@pglossarysection[2]{%
  \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@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
  \newcommand*{\@xdycounters}{\glscounter}
  \newcommand*{\GlsAddXdyCounters}[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
    }%
  }%

```

```

        \@do@addcounter
    }
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
    \glsnxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glssaddxdycounters{%
    \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
\newcommand*\@glssaddxdyattribute[2]{%
    \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
    \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
    \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
    \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
    \fi
    \@for\@this@counter:=\@xdycounters\do{%
        \protected@edef\gls@do@addxdyattribute{%
            \noexpand\@glssaddxdyattribute{#1}{\@this@counter}%
        }
        \gls@do@addxdyattribute
    }%
    \let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
    \glsnxindywarning\GlsAddXdyAttribute}
\fi

```



```

\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{glshypernumber}
  \GlsAddXdyAttribute{hyperrm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^~J
      (define-alphabet "#1" (#2))}%
  }
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"

```

```

\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@gls@Alphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\@GlsAddXdyLocation}[3][[]]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^~J%

```

```

        (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glsoopenbrace\string" #3
        :sep \string"\glsclosebrace\string"))
    }%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glsoopenbrace"
#1
:sep "\glsclosebrace\glsoopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
\expandafter\toks@\expandafter{\@xdysortrules}%
\protected@edef\@xdysortrules{\the\toks@ ^^J

```

```

        (sort-rule \string"#1\string" \string"#2\string"))}%
    }
\else
    \newcommand*\GlsAddSortRule[2]{%
        \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
    \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
    \newcommand*\GlsAddXdyStyle[1]{%
        \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
    \newcommand*\GlsAddXdyStyle[1]{%
        \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
    \newcommand*\GlsSetXdyStyles[1]{%
        \edef\@xdyrequiredstyles{#1}}
\else
    \newcommand*\GlsSetXdyStyles[1]{%
        \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{}
\def\@xdylanguage#1#2{}
\ifglxindy
    \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
        \ifglossaryexists{#1}{%
            \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
        }{%
            \PackageError{glossaries}{Can't set language type for
                glossary type '#1' --- no such glossary}{%
                You have specified a glossary type that doesn't exist}}
\else
    \newcommand*\GlsSetXdyLanguage[2][{}]{%
        \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
    \newcommand*\GlsSetXdyCodePage[1]{%
        \renewcommand*\@gls@codepage{#1}%
    }
    \AtBeginDocument{%
        \ifx\@gls@codepage\@empty
            \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
        \fi
    }
\else
    \newcommand*\GlsSetXdyCodePage[1]{%
        \glsnoxywarning\GlsSetXdyCodePage}

```

```

\fi
\ifglxsindy
\ifglxsindy@glslnumbers
\def\@xdylettergroups{(define-letter-group
\string"glslnumbers\string"^^J\space\space\space
:prefixes (\string"0\string" \string"1\string"
\string"2\string" \string"3\string" \string"4\string"
\string"5\string" \string"6\string" \string"7\string"
\string"8\string" \string"9\string")^^J\space\space\space
\@xdynumbergrouporder)}
\else
\def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
\expandafter\toks@\expandafter{\@xdylettergroups}%
\protected@edef\@xdylettergroups{\the\toks@^^J%
(define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
\@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
\@for#1:=\@glslacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\@glsldefaulttype]{%
\edef\@glo@list{\csname glolist@#1\endcsname}%
\@for#2:=\@glo@list\do
{%
\ifdefempty{#2}{\fi}%
}%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
\expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
{%
\forallglsentries[\@this@glo@]{#2}{#3}%
}%
}
\newcommand*\ifglossaryexists{%
\@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand*\@ifglossaryexists[3]{%
\ifcsundef{\@glo@type@#1@out}{#3}{#2}%
}
\newcommand*\s@ifglossaryexists[3]{%
\ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glslsetoklabel[1]{#1}
\newcommand*\ifglsentryexists[3]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@glseentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs\glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
        {%

```

```

        \def\do@glshaschildren{#2}%
        \@endfortrue
    }%
    {}%
}%
\do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
    \glstoifexists{#1}%
    {%
        \ifcsepty{glo@\glstetoklabel{#1}@parent}{#3}{#2}%
    }%
}
\newcommand*{\ifglshasdesc}[3]{%
    \ifcsepty{glo@\glstetoklabel{#1}@desc}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glstetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
}
}
\newrobustcmd*{\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glstetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
        \ifdefequal\@glo@symbol\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glstetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
        \ifdefequal\@glo@long\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glstetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
        \ifdefequal\@glo@short\@gls@default@value

```

```

        {#3}%
        {#2}%
    }%
}
\newrobustcmd*{\ifglshasfield}[4]{%
    \glsoifexists{#2}%
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
        \ifdef\@glo@thisvalue
        {%
            \ifdefempty\@glo@thisvalue
            {%
                #4%
            }%
        }%
        {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
                #4%
            }%
            {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
            }%
        }%
    }%
}%
{%
    \@gls@fetchfield{\@gls@thisfield}{#1}%
    \ifdef\@gls@thisfield
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@ \@gls@thisfield}%
        \ifdefempty\@glo@thisvalue
        {%
            #4%
        }%
        {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
                #4%
            }%
            {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
            }%
        }%
    }%
}%
{%
    \GlossariesWarning{Unknown entry field '#1'}%
    #4%
}%
}%

```



```

}%
}
\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary{%
  \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*{\defglsentryfmt}[2][\glsdefaultttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtextext}[1]{%
  \ifboolexpr{test \ifdefstring{#1}{tex}}
    or test \ifdefstring{#1}{TEX}}
  {%
    \def#1{nottex}%
    \PackageError{glossaries}%
      {Forbidden '.tex' extension replaced with '.nottex'}%
      {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
        Don't use '.tex' as an extension for a temporary file.}%
  }%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[\@gls@gobbleopt]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}{\@ifstar\s@newglossary\@ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaultttype
    {%
      \gdef\glsdefaultttype{#2}%
    }%
  }%
  \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
  \expandafter\gdef\csname glolist@#2\endcsname{,}%
  \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
  \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
  \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
  \expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
  \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
  \@gls@provide@newglossary
  \protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}%
  \ifcsundef{gls@#2@entryfmt}%

```

```

{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxsindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \edef\@ignored@glossaries{#1}%
  }%
  {%
    \eappto\@ignored@glossaries{,#1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%

```

```

    {%
      \eappto\@gls@nohyperlist{, #1}%
    }%
  }
  \newcommand*{\@ignored@glossaries}{}
  \newcommand*{\ifignoredglossary}[3]{%
    \edef\@gls@igtype{#1}%
    \expandafter\DTLifinlist\expandafter
      {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
  }
  \define@key{glossentry}{name}{%
    \def\@glo@name{#1}%
  }
  \define@key{glossentry}{description}{%
    \def\@glo@desc{#1}%
  }
  \define@key{glossentry}{descriptionplural}{%
    \def\@glo@descplural{#1}%
  }
  \define@key{glossentry}{sort}{%
    \def\@glo@sort{#1}}
  \define@key{glossentry}{text}{%
    \def\@glo@text{#1}%
  }
  \define@key{glossentry}{plural}{%
    \def\@glo@plural{#1}%
  }
  \define@key{glossentry}{first}{%
    \def\@glo@first{#1}%
  }
  \define@key{glossentry}{firstplural}{%
    \def\@glo@firstplural{#1}%
  }
  \newcommand*{\@gls@default@value}{\relax}
  \define@key{glossentry}{symbol}{%
    \def\@glo@symbol{#1}%
  }
  \define@key{glossentry}{symbolplural}{%
    \def\@glo@symbolplural{#1}%
  }
  \define@key{glossentry}{type}{%
    \def\@glo@type{#1}}
  \define@key{glossentry}{counter}{%
    \ifcsundef{c@#1}%
    {%
      \PackageError{glossaries}%
        {There is no counter called ‘#1’}%
      {%
        The counter key should have the name of a valid counter
        as its value%
      }
    }
  }

```

```

    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning{glossaries}%
  {'\gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
[\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*{\@gls@savenonumberlist}[1]{}
\newcommand*{\@gls@initnonumberlist}{}%
\newcommand*{\@gls@storenonumberlist}[1]{}
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%

```

```

\ifdef\@glo@nonumberlist
{%
  \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
}%
{}%
}%
\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*{\@glsnodelsc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}

```

```

}%
}%
\newcommand*{\@glsdefaultplural}{%
\newcommand*{\@gls@missingnumberlist}[1]{%
??%
\ifglssavenumberlist
\GlossariesWarning{Missing number list for entry '#1'.
Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option 'savenumberlist=true' required}%
{%
You must use the 'savenumberlist' package option
to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}

```

```

\newcommand{\@gls@expand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
      {%
        \@gls@expand@field{#2}{#3}{#4}%
      }%
      {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
      }%
    }%
  }%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\gls@defdocnewglossaryentry}{%
  \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
  \let\newglossaryentry\new@glossaryentry
}

```

```

\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglentryexists{#1}{}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
  \@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*{\gls@begindocdefs}{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{\relax}{\relax}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglentryexists{#1}{\glspercentchar^^J%
    \expandafter\@gobble\string{\glspercentchar^^J%
      \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
      \expandafter\@gobble\string{\glspercentchar%
    }%
    \@for\@gls@map:=\@gls@keymap\do
    {%
      \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
      \ifdef\glo@value
      {%
        \@onelevel@sanitize\glo@value
        \immediate\write\@gls@deffile
        {%
          \expandafter\@firstoftwo\@gls@map
          =\expandafter\@gobble\string{\glo@value\expandafter\@gobble\string\},%
        }%
      }%
    }%
  }%
}

```



```

        \glspercentchar
    }%
}%
{}%
}%
\glswritedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar%
}%
}
\newcommand*{\@gls@keymap}{%
    {name}{name},%
    {sort}{sortvalue},% unescaped sort value
    {type}{type},%
    {first}{first},%
    {firstplural}{firstpl},%
    {text}{text},%
    {plural}{plural},%
    {description}{desc},%
    {descriptionplural}{descplural},%
    {symbol}{symbol},%
    {symbolplural}{symbolplural},%
    {user1}{useri},%
    {user2}{userii},%
    {user3}{useriii},%
    {user4}{useriv},%
    {user5}{userv},%
    {user6}{uservi},%
    {long}{long},%
    {longplural}{longpl},%
    {short}{short},%
    {shortplural}{shortpl},%
    {counter}{counter},%
    {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
    \edef\@gls@thisval{#2}%
    \@for\@gls@map:=\@gls@keymap\do{%
        \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
        \ifdefequal{\@this@key}{\@gls@thisval}%
        {%
            \edef#1{\expandafter\@secondoftwo\@gls@map}%
            \@endfortrue
        }%
    }%
}%
}

```

```

\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
}%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
  }%
}

```

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  {}%
}%
{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1@\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2} []}%
    \csdef{@gls@user@#1@}##1##2[##3]{%
      \@gls@field@link{##1}{##2}{#3{##2}##3}%
    }%
    \newrobustcmd*{#5}{%
      \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
    }%
  \ifcsdef{@Gls@user@#1@}%
  {%
    \PackageError{glossaries}%
    {Can't define '\string#6' as helper command
     '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
    {}%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname @Gls@user@#1@\endcsname}[2] []{%
      \new@ifnextchar[%
        {\csuse{@Gls@user@#1@}{##1}{##2}}%
        {\csuse{@Gls@user@#1@}{##1}{##2} []}%
      \csdef{@Gls@user@#1@}##1##2[##3]{%
        \@gls@field@link{##1}{##2}{#4{##2}##3}%
      }%
      \newrobustcmd*{#6}{%
        \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
      }%
    \ifcsdef{@GLS@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can't define '\string#7' as helper command
       '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
      {}%
    }%
    {%
      \expandafter\newcommand\expandafter*\expandafter
      {\csname @GLS@user@#1@\endcsname}[2] []{%

```

```

\new@ifnextchar[%
  {\csuse{@GLS@user@#1@}{##1}{##2}}%
  {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand{\glsfieldxddef}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csxdef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\glsfielddedef}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \protected@csedef{glo@\@glo@label @#2}{#3}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{%
  \edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}

```

```

    }%
  }%
}
\newcommand{\glsfielddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \letcs#3{glo@\@glo@label @#2}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\ifglsfieldeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\ifglsfielddefeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\ifdefstrequal
        \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
    }%
  }%
}

```

```

}%
{%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
        \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
}%
}
\newcommand*{\glswritedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
    \gls@assign@field{}{#1}{desc}{\@glo@desc}%
    \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
    \glsdoifnoexists{#1}%
    {%
        \bgroup
        \let\@org@newglossaryentryprehook\@newglossaryentryprehook
        \long\def\@newglossaryentryprehook{%
            \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
            \@org@newglossaryentryprehook
        }%
        \renewcommand*{\gls@assign@desc}[1]{%
            \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
            \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
        }
        \gls@defglossaryentry{#1}{#2}%
        \egroup
    }
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
    \ifglsentryexists{#1}{}%
    {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
    \let\GlsSetQuote\gls@nosetquote
    \edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\let\glslabel\@glo@label
\let\@glo@name\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type
\let\@glo@symbol\@gls@symbol
\let\@glo@symbolplural\@gls@symbolplural
\let\@glo@text\@gls@text
\let\@glo@plural\@gls@plural
\let\@glo@first\@gls@first
\let\@glo@firstplural\@gls@firstplural
\let\@glo@sort\@gls@sort
\let\@glo@counter\@gls@counter
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonnumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{%
  \ifignoredglossary\@glo@type
  {%
    \ifx\@glo@desc\@gls@desc

```

```

        \let\@glo@desc\@empty
    \fi
}%
{%
}%
\protected@edef\@glolist@\{ \csname glolist@\@glo@type\endcsname\}%
\expandafter\xdef\csname glolist@\@glo@type\endcsname{%
    \@glolist@\{ \@glo@label\},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
\else
    \ifdefequal\@glo@label\@glo@parent%
    {%
        \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
        \def\@glo@parent{%
        \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
        }%
        {%
            \ifglsentryexists{\@glo@parent}%
            {%
                \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
                    \@glo@parent}%
                \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
                \advance\gls@level by 1\relax
                \ifx\@glo@name\@glsnoname
                    \expandafter\let\expandafter\@glo@name
                    \csname glo@\@glo@parent @name\endcsname
                    \ifx\@glo@plural\@gls@default@value
                        \expandafter\let\expandafter\@glo@plural
                        \csname glo@\@glo@parent @plural\endcsname
                    \fi
                \fi
            }%
        }%
        {%
            \PackageError{glossaries}%
            {%
                Invalid parent ‘\@glo@parent’
                for entry ‘\@glo@label’ - parent doesn’t exist%
            }%
            {%
                Parent entries must be defined before their children%
            }%
            \def\@glo@parent{%
            \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
            }%
            }%
        }%
    \fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%

```



```

\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{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@ddesc{\@glo@first}%
  \ifx\@glo@desc\@glo@ddesc

```

```

\let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodedesc
\@glsnodedesc
\let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
\let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
\csname glo@\@glo@label @symbol\endcsname{
\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\@do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%

```

```

\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\@glo@list{,%}
  \for@gl@entries[\@glo@type]{\@glo@label}%
  {%
    \if@defequal\@glo@thislabel\@glo@label
      {}{\@eappto\@glo@list{\@glo@label},}%
    }%
  \cslet\@glolist@\@glo@type{\@glo@list}%
  \csdef\@glo@\@glo@thislabel @type#{#2}%
}
\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}%
  \@gl@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gl@checkmkidxchars\@glo@sort
  \@gl@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@glstetoklabel{#1}@flag\endcsname\iftrue
}%
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
    \expandafter\global\csname glo@glstetoklabel{#1}@flagtrue\endcsname
}%
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
    \expandafter\let\csname ifglo@glstetoklabel{#1}@flag\endcsname\iffalse
}%
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
    \expandafter\global\csname glo@glstetoklabel{#1}@flagfalse\endcsname
}%
\newcommand*{\glsresetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glsreset{\@glsentry}%
    }%
}%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glslocalreset{\@glsentry}%
    }%
}%

```

```

}%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*{\glsenableentrycount}{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*{\newglossaryentry}[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*{\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*{\@glslocalreset}[1]{%

```



```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
    }%
  }%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][{}]{%
  \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}}[{}]}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
    \string\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}
\newcommand*{\@cGls}[2][{}]{%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}}[{}]}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
  \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}}[{}]}%
}
\def\@cglspl@#1#2[#3]{%
  \GlossariesWarning{\string\cglspl\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
  \@glspl@{#1}{#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}}[{}]}%
}

```



```

}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2} [#3]%
}
\newcommand*\cGlsplformat[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glstextformat[1]{#1}
\newcommand*\glsentryfmt{%
  \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*\@@gls@default@entryfmt[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \ifbool{glscompatible-3.07}%
        {%
          \protected@edef\@glo@etext{%
            #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }
          \xmakefirstuc\@glo@etext
        }%
        {%

```

```

        #2{\Glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
}%
{%
    \ifbool{glscompatible-3.07}%
    {%
        \protected@edef\@glo@etext{%
            #1{\Glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
        \xmakefirstuc\@glo@etext
    }%
    {%
        #1{\Glsentryfirstplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
}%
}%
{%
    \ifglsused\glslabel
    {%
        \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    }%
    {%
        \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    }%
}%
}%
{%
    \glscapscase
    {%
        \ifglsused\glslabel
        {%
            #2{\glsentrytext{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}%
        }%
        {%
            #1{\glsentryfirst{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}%
        }%
    }%
}%

```

```

{%
  \ifglssused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
  }%
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #1{\Glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
  }%
  {%
    \ifglssused\glslabel
    {%
      \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
    {%
      \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
  }%
  {%
    \ifglssused{\glslabel}%

```

```

    {%
      #2{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{}}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase

```

```

{%
  \ifglused\glslabel
  {%
    \glentrytext{\glslabel}\glsinsert
  }%
  {%
    \glentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \mfirstucMakeUppercase{\glentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase{\glentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```

```

    }%
}%
{%
  \glscapscase
  {%
    \acronymfont{\glsentryshort{\glslabel}}\glsinsert
  }%
  {%
    \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
  }%
  {%
    \mfirstucMakeUppercase
    {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
  }%
}%
}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}%
{%
  \glscapscase
  {%
    \genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\genacrfullformat{\glslabel}{\glsinsert}}%
  }%
}%
}%
}%
{%
  \glscustomtext
}%
}

```

```

\newcommand*{\genacrfullformat}[2]{%
  \glentrylong{#1}#2\space
  (\protect\firstacronymfont{\glentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\genplacrfullformat}[2]{%
  \glentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glentryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\glsdisplayfirst}[4]{#1#4}
\newcommand*{\glsdisplay}[4]{#1#4}
\newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
    {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
    }%
    }%
    }%
    \@gls@doentrydef
}
\define@key{glslink}{counter}{%
    \ifcsundef{c@#1}%
    {%
        \PackageError{glossaries}%
        {There is no counter called ‘#1’}%
        {%
            The counter key should have the name of a valid counter
            as its value%
        }%
    }%
    }%
    {%
        \def\@gls@counter{#1}%
    }%
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
    \glslinkvar{#1}{#2}{#1}%
    \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
        you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
    \@gls@hyp@opt\@gls@link
}

```



```

\newcommand*{\@gls@link}[3][\{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{\glsnumberformat}%
  \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
}

```

```

\glslinkpostsetkeys
\@gls@saveentrycounter
\@gls@setsort{\glslabel}%
\@do@wrglossary{#2}%
\ifKV@glslink@hyper
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}{%
    {
      \ifcsundef{xatlevel@}%
      {%
        \edef\theglsentrycounter{\expandafter\noexpand
          \csname the\@gls@counter\endcsname}%
      }%
      {%
        \ifx\xatlevel@\@empty
          \edef\theglsentrycounter{\expandafter\noexpand
            \csname the\@gls@counter\endcsname}%
        \else
          \savecounters@
          \advance\c@equation by 1\relax
          \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
          \ifcsundef{theH\@gls@counter}%
          {%
            \def\@gls@Hcounter{\theglsentrycounter}%
          }%
          {%
            \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
          }%
          \protected@edef\theHglentrycounter{\@gls@Hcounter}%
          \restorecounters@
        \fi
      }%
    }%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    \ifx\@gls@Hcounter\@empty
      \ifcsundef{theH\@gls@counter}%
      {%
        \def\theHglentrycounter{\theglsentrycounter}%
      }%
    }%
  }%
}

```

```

}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1\relax
\def\@glo@prefix{()}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%

```

```

\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostsubst{%
  \noexpand\DTLsubstituteall\noexpand\gls@xdystring
  {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil""\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil"??\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                }
              }
            }
          }
        }
      }
    }
  \fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\@nil#2\@nil
    \ifx\@nil#3\@nil
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkquote{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
      \def\@gls@checkquote{\@gls@checkquote#3\@nil}%
    \fi
  }

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\@null#3\@null
\def\@gls@checkquote{\@gls@checkquote#2""\@null}%
\else
\def\@gls@checkquote{\@gls@checkquote#2"#3\@null}%
\fi
\fi
\@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\@null#3\@null
\def\@gls@checkescquote{\@gls@checkescquote#2""\@null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2"#3\@null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

\@gls@quotechar\string\" \@gls@actualchar}%
\ifx\@null#3\@null
\def\@gls@checkescactual{\@gls@checkescactual#2\? \? \@null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\@null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@encapchar
\@gls@quotechar\string\" \@gls@encapchar}%
\def\@gls@checkescbar{\@gls@checkescbar#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@encapchar}%
\ifx\@null#3\@null
\def\@gls@checkescbar{\@gls@checkescbar#2\|\|\@null}%
\else
\def\@gls@checkescbar{\@gls@checkescbar#2\|#3\@null}%
\fi
\fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesclevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@levelchar
\@gls@quotechar\string\" \@gls@levelchar}%
\def\@gls@checkesclevel{\@gls@checkesclevel#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@levelchar}%
\ifx\@null#3\@null

```

```

\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!\!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi

```

```

\fi
\@@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@@gls@checkactual
}
\def\@gls@xdycheckquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@xdycheckquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"\string"}%
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"}%
\ifx\null#3\null
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\@@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%

```



```

\newcommand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\newcommand\null{%
\newcommand\@gls@tmpb=\newcommand\expandafter
{\newcommand\@gls@checkedmkidx}%
\newcommand\toks@={##1}%
\newcommand\ifx\newcommand\null##2\newcommand\null
\newcommand\ifx\newcommand\null##3\newcommand\null
\newcommand\edef\newcommand\@gls@checkedmkidx{%
\newcommand\the\newcommand\@gls@tmpb\newcommand\the\newcommand\toks@}%
\newcommand\def\newcommand\@gls@xdycheckbackslash{\relax}%
\newcommand\else
\newcommand\edef\newcommand\@gls@checkedmkidx{%
\newcommand\the\newcommand\@gls@tmpb\newcommand\the\newcommand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\newcommand\def\newcommand\@gls@xdycheckbackslash{%
\newcommand\@gls@xdycheckbackslash##3\newcommand\null}%
\newcommand\fi
\newcommand\else
\newcommand\edef\newcommand\@gls@checkedmkidx{%
\newcommand\the\newcommand\@gls@tmpb\newcommand\the\newcommand\toks@
\@backslashchar\@backslashchar}%
\newcommand\ifx\newcommand\null##3\newcommand\null
\newcommand\def\newcommand\@gls@xdycheckbackslash{%
\newcommand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\newcommand\null}%
\newcommand\else
\newcommand\def\newcommand\@gls@xdycheckbackslash{%
\newcommand\@gls@xdycheckbackslash##2\@backslashchar
##3\newcommand\null}%
\newcommand\fi
\newcommand\fi
\newcommand\@gls@xdycheckbackslash
}%
}
\def\@gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*\@glsdohypertarget}[2]{%
\@glsshowtarget{#1}%
\settoheight{\gls@tmplen}{#2}%
\raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*\@glsdohyperlink}[2]{%
\@glsshowtarget{#1}%
\hyperlink{#1}{#2}%
}
\newcommand*\@glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
\let\@glslink\@glsdonohyperlink

```

```

}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
}
\def\@Gls@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\gl@sifplural\@secondoftwo
\let\gl@scapscase\@secondofthree
\let\gl@customtext\@empty
\def\gl@insert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link{#1}{#2}{\@glo@text}%
\ifKV@gl@link@local
\gl@localunset{#2}%
\else
\gl@unset{#2}%
\fi
}%
\gl@postlinkhook
}
\newrobustcmd*{\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\gl@sifplural\@secondoftwo
\let\gl@scapscase\@thirdofthree
\let\gl@customtext\@empty
\def\gl@insert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link{#1}{#2}{\@glo@text}%
\ifKV@gl@link@local
\gl@localunset{#2}%
\else
\gl@unset{#2}%
\fi
}%
\gl@postlinkhook
}
\newrobustcmd*{\gl@spl}{\@gl@hyp@opt\@gl@spl}
\newcommand*{\@gl@spl}[2][{}]{%
\new@ifnextchar[{\@gl@spl@{#1}{#2}}{\@gl@spl@{#1}{#2}[]}%
}
\def\@gl@spl@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\gl@sifplural\@firstoftwo
\let\gl@scapscase\@firstofthree
\let\gl@customtext\@empty

```

```

\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\newcommand*{\@GLSpl}[2][]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```

```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][\@glsdoifexists{#2}]{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsupcase\@firstofthree
    \def\glsentrytext{#3}%
    \def\glsentry{#2}%
    \def\gls@link@csname{gls@glstype @entryfmt\endcsname}%
    \@gls@link{#1}{#2}{\@gls@link@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*{\glsentrytext}{\@gls@hyp@opt\@glsentrytext}
\newcommand*{\@glsentrytext}[2][\@glsentrytext@{#1}{#2}]{%
    \new@ifnextchar[{\@glsentrytext@{#1}{#2}}{\@glsentrytext@{#1}{#2}[]}}
\def\@glsentrytext@{#1}{#2}[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][\@GLStext@{#1}{#2}]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
\def\@GLStext@{#1}{#2}[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\Glsentrytext}{\@gls@hyp@opt\@Glsentrytext}
\newcommand*{\@Glsentrytext}[2][\@Glsentrytext@{#1}{#2}]{%
    \new@ifnextchar[{\@Glsentrytext@{#1}{#2}}{\@Glsentrytext@{#1}{#2}[]}}
\def\@Glsentrytext@{#1}{#2}[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}

```

```

\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][]{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}]
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][]{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][]{%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]}]
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2][]{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2}[]}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2][]{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2][]{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2}[]}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][]{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][]{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2][]{%

```

```

\new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}]
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\@glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\@GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@GLSentryname{#2}#3}%
}
\newrobustcmd*{\@GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\@glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@GLSentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\@glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\@GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] []{%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def\@GLSdescplural@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glssymbol}{\@gls@hyp@opt\@Glssymbol}
\newcommand*{\@Glssymbol}[2] [] {%
  \new@ifnextchar[{\@Glssymbol@{#1}{#2}}{\@Glssymbol@{#1}{#2} []}}
\def\@Glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
  \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
  \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}}
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glssymbolplural}{\@gls@hyp@opt\@Glssymbolplural}
\newcommand*{\@Glssymbolplural}[2] [] {%
  \new@ifnextchar[{\@Glssymbolplural@{#1}{#2}}{\@Glssymbolplural@{#1}{#2} []}}
\def\@Glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}

```



```

\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][]{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2][]{%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2][]{%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][]{%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][]{%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\newcommand*{\@GLSuseriii}[2][]{%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2][]{%

```

```

\new@ifnextchar [{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} [] }}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar [{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} [] }}
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} [] }}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} [] }}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] [] {%
  \new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} [] }}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} [] }}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} [] }}
\def\@GLSuservi@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}{#3}}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2][]{%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2}[]}%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2][]{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][]{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}%
}

```

```

\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glstryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@acrlong{\@gl@hyp@opt\@ns@acrlong}
\newcommand*\@ns@acrlong[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glstrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@Acrlong{\@gl@hyp@opt\@ns@Acrlong}
\newcommand*\@ns@Acrlong[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2][]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%

```

```

\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
  }%
  ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
  entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%
{%

```

```

    ??%
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  \else
    \expandafter\ifx\@gls@body\glsentryshort\relax
    \expandafter\Glsentryshort\@gls@rest
  \else
    \expandafter\ifx\@gls@body\acronymfont\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
    {%
      \let\glsentryshort\Glsentryshort
      \@glo@text
    }%
  \else
    \xmakefirstuc{\@glo@text}%
  \fi
  \fi
  \fi
}
}%
{%
  \@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*\glentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*\Glsentrytext}[1]{%
  \@Gls@entry@field{#1}{text}%
}
\newcommand*\glentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*\Glsentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newcommand*\glentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*\Glsentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newcommand*\glentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*\glentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newrobustcmd*\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*\glentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*\@glentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@Gls@value}%
    \xcapitalisewords{\@Gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*\glentrytitlecase}[2]{%
    \texorpdfstring
      {\@glentrytitlecase{#1}{#2}}%
      {\@Gls@entry@field{#1}{#2}}%
  }
}

```

```

    }
}
{
    \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glsentrysort}[1]{%
    \@gls@entry@field{#1}{sort}%
}
\newcommand*{\glsentryparent}[1]{%
    \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glsentryuseri}[1]{%
    \@gls@entry@field{#1}{useri}%
}
\newrobustcmd*{\Glsentryuseri}[1]{%
    \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\glsentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*{\Glsentryuserii}[1]{%
    \@Gls@entry@field{#1}{userii}%
}
\newcommand*{\glsentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*{\Glsentryuseriii}[1]{%
    \@Gls@entry@field{#1}{useriii}%
}
\newcommand*{\glsentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*{\Glsentryuseriv}[1]{%
    \@Gls@entry@field{#1}{useriv}%
}
\newcommand*{\glsentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*{\Glsentryuserv}[1]{%
    \@Gls@entry@field{#1}{userv}%
}
\newcommand*{\glsentryservi}[1]{%
    \@gls@entry@field{#1}{servi}%
}
\newrobustcmd*{\Glsentryservi}[1]{%
    \@Gls@entry@field{#1}{servi}%
}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*{\Glsentryshort}[1]{%

```

```

    \@Gls@entry@field{#1}{short}%
}
\newcommand*{\glsentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}
\newrobustcmd*{\Glsentryshortpl}[1]{%
    \@Gls@entry@field{#1}{shortpl}%
}
\newcommand*{\glsentrylong}[1]{\@Gls@entry@field{#1}{long}}
\newrobustcmd*{\Glsentrylong}[1]{%
    \@Gls@entry@field{#1}{long}%
}
\newcommand*{\glsentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
\newrobustcmd*{\Glsentrylongpl}[1]{%
    \@Gls@entry@field{#1}{longpl}%
}
\newcommand*{\glsentryfull}[1]{%
    \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
    \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
    \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
    \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
        \@Gls@entry@field{#1}{numberlist}%
    }%
}
\@ifpackageloaded{hyperref} {%
    \newcommand*{\glsdisplaynumberlist}[1]{%
        \GlossariesWarning
        {%
            \string\glsdisplaynumberlist\space
            doesn't work with hyperref.^^JUsing
            \string\glsentrynumberlist\space instead%
        }%
        \glsentrynumberlist{#1}%
    }%
}%
{\%
    \newcommand*{\glsdisplaynumberlist}[1]{%
        \glsdoifexists{#1}%
        {%
            \bgroup
            \edef\@glo@label{\glsdetoklabel{#1}}%
            \let\@org@glsnumberformat\glsnumberformat

```

```

\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
  \csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][ ]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{\glsnumberformat}%
    \edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*{\glsaddall}[1][ ]{%
  \edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*{\glsignore}[1]{%
  \edef\glsopenbrace{\expandafter\@gobble\string\{ }
  \edef\glsclosebrace{\expandafter\@gobble\string\} }
  \edef\glsbackslash{\expandafter\@gobble\string\ }
  \edef\glsquote#1{\string"#1\string"}
  \edef\glspercentchar{\expandafter\@gobble\string\% }
  \edef\glsstildechar{\string~}
  \ifglsxindy
    \newcommand*{\@glsfirstletter}{A}
  \fi
  \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
    \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
  \ifglsxindy
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}%
      \@onelevel@sanitize\@glsfirstletter
    }
  \else
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \glsnoindywarning\GlsSetXdyFirstLetterAfterDigits}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
      \@GlsSetXdyFirstLetterAfterDigits
    }
  \fi
  \ifglsxindy
    \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
  \fi
  \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
  }
  \ifglsxindy
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
      \renewcommand*{\@xdynumbergrouporder}{#1}%
    }
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

\renewcommand*{\@xdynumbergrouporder}{#1}%
\@onelevel@sanitize\@xdynumbergrouporder
}
\else
\newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
\glsnoxywarning\GlsSetXdyNumberGroupOrder}
\newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
\@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsmminrange}{2}
\ifglsxindy
\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
\renewcommand*{\@glsmminrange}{#1}}
\else
\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\@glscataddpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{ }(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@glsc@classI:=\@glsc@xdy@locationlist\do{%
\protected@write\glswrite{ }(define-location-class
\string"\@glsc@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @glsc@xdy@Lclass@\@glsc@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```

```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}\{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "{{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glsseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string

```

```

\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glresetentrylist
\glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
\string"\glsoopenbrace\string\glossaryentrynumbers
\glsoopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsoopenbrace\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{\glspcrpercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspcrpercentchar\space for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}

```

```

\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glresetentrylist \string}
\write\glswrite{symhead_positive \string\glssymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string\gls@suffixFF\string}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
  \@ifpackageloaded{tracklang}%
  {%
    \IfTrackedLanguage{german}%
    {%
      \def\@gls@extramakeindexopts{-g}%
    }%
  }%
}%
\edef\@gls@docheckquotedef{%
  \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
    \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
    \noexpand\toks@={####1}%

```

```

\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%

```

```

\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
{\string\GlsSetQuote\space not permitted here}%
{Move \string\GlsSetQuote\space earlier in the preamble, as
soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*{\@gls@extramakeindexopts}[1]{%
\newcommand{\noist}{%
\@gls@addpredefinedattributes
\let\writeist\relax
}
\newcommand*{\@makeglossary}[1]{%
\ifglossaryexists{#1}%
{%
\ifglssavewrites
\expandafter\newtoks\csname glo@#1@filetok\endcsname
\else
\expandafter\newwrite\csname glo@#1@file\endcsname
\expandafter\@gls@openfile\csname glo@#1@file\endcsname{#1}%
\fi
\@gls@renewglossary
\writeist
}%
{%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@gls@openfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglxsindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}

```

```

{%
  \@closegls{#1}%
  \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\empty}%
  \ifcsundef{@xdy@#1@language}%
    {\let\@gls@langmod\@xdy@main@language}%
    {\letcs\@gls@langmod{@xdy@#1@language}}%
  \edef\@gls@dothiswrite{\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \@gls@istfilebase\space
    -C \@gls@codepage\space
    -t \jobname.\csuse{@glotype@#1@log}
    -o \jobname.\csuse{@glotype@#1@in}
    \jobname.\csuse{@glotype@#1@out}}}%
 }%
  \@gls@dothiswrite
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
      {\def\@gls@order{-l }}%
      {\let\@gls@order\empty}%
    \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
      -s \istfilename\space
      -t \jobname.\csuse{@glotype@#1@log}
      -o \jobname.\csuse{@glotype@#1@in}
      \jobname.\csuse{@glotype@#1@out}}}%
   }%
    \@gls@dothiswrite
 }%
  {\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\fi
\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%

```

```

{%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
  \ifcsundef{xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{xdy@#1@language}}}%
  \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \gls@istfilebase\space
    -C \gls@codepage\space
    -t \jobname.\csuse{\@glotype@#1@log}
    -o \jobname.\csuse{\@glotype@#1@in}
    \jobname.\csuse{\@glotype@#1@out}}}%
  }%
  \@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}
  doesn't exist. Rerun may be required}}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{\@glotype@#1@out}}{%
      {%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-l }}%
        {\let\@gls@order\@empty}%
        \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
          -s \istfilename\space
          -t \jobname.\csuse{\@glotype@#1@log}
          -o \jobname.\csuse{\@glotype@#1@in}
          \jobname.\csuse{\@glotype@#1@out}}}%
        }%
        \@gls@dothiswrite
      }%
      {\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}
        doesn't exist. Rerun may be required}}}%
    }%
    {\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
  }%
}
}

```

```

\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}%
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}%
\ifundef\@gls@extramakeindexopts
{}%
}%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}%
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}%
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{} }{}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\def\warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
}%

```



```

\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
\@domakeglossaries
{%
\renewcommand{\@gls@noref@warn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
\let\@gls@checkmkidxchars\@gobble
\glsesclocationsfalse
\let\@do@wrglossary\gls@noidxglossary
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
\edef\@gls@label{\glsdetoklabel{##1}}}%
\protected@write\@auxout{}{%
\string\@gls@reference

```

```

        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}%
        {%
            \string\glsseeformat##2}%
        }%
    }%
}%
\AtBeginDocument
{%
    \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
    \GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@assign@sortkey
    \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*{\new@glossaryentry[2]{%
    \PackageError{glossaries}{Glossary entries must be
defined in the preamble^^Jwhen you use
\string\makenoidxglossaries}%
    {Either move your definitions to the preamble or use
\string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
    \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
    \ifdef\@gls@loclist
    {%
        \glsnoidxloclist{\@gls@loclist}%
    }%
    {%
        ??\glsdoifexists{##1}%
        {%
            \GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven’t referenced the entry}%
        }%
    }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
    \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
    \ifdef\@gls@loclist
    {%
        \def\@gls@noidxloclist@sep{%
            \def\@gls@noidxloclist@sep{%

```

```

\def\@gls@noidxloclist@sep{%
  \glsnumlistsep
}%
\def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
}%
}%
\def\@gls@noidxloclist@finalsep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
\@gls@noidxloclist@finalsep
\@gls@noidxloclist@prev
}%
{%
??\glsdoifexists{##1}%
{%
  \GlossariesWarning{Missing location list for ‘##1’. Either
    a rerun is required or you haven’t referenced the entry}%
}%
}%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org\glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org\glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org\glsnoidxdisplayloc
  \let\glsseeformat\@gls@org\glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%

```

```

}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
    {\@glo@counterprefix}%
    {\@gls@counter}%
    {\@glsnumberformat}%
    {\@glslocref}%
    }%
  }%
}
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \fi
  }%
}

```

```

        \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp~J}%
    \else
        \ifcsdef{glo@#1@file}%
        {%
            \expandafter\protected@write\csname glo@#1@file\endcsname{%
                \gls@disablepagerefexpansion}{#2}%
            }%
        {%
            \ifignoredglossary{#1}{}%
            {%
                \GlossariesWarning{No file defined for glossary ‘#1’}%
            }%
        }%
    \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*{\glswriteentry}[2]{%
    \ifglsexonlyfirst
        \ifglssused{#1}{#2}%
    \else
        #2%
    \fi
}

```

```

\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\glsaddprotectedpagefmt}[1]{%
\ea\pt\o\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
\csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
\ea\pt\o\@wrglossarynumberhook{%
\noexpand\let\expandonce{\csname org@gls#1\endcsname}%
\expandonce{\csname#1\endcsname}%
\noexpand\def\expandonce{\csname#1\endcsname}{%
\noexpand\@wrglossary@pageformat
\expandonce{\csname gls#1page\endcsname}%
\expandonce{\csname org@gls#1\endcsname}%
}%
}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
\ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@do@wrglossary}[1]{%
\ifglsclocations
\@do@esc@wrglossary{#1}%
\else
\@do@noesc@wrglossary{#1}%
\fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{}%
\else
\protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix

```

```

\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@@do@wrglossary

```

```

}
\newcommand*{\@@do@@wrglossary}{%
  \ifglxsindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \gls@glossary{csname glo@\@gls@label @type\endcsname}{%
    (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
      :locref \string{\@glo@counterprefix}{\@glslocref}\string" %
      :attr \string"@gls@counter\@glo@suffix\string"
      \@glo@range
    )
  }%
\else
  \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
  {\@glo@counterprefix}%
  \gls@glossary{csname glo@\@gls@label @type\endcsname}{%
    \string@glossaryentry{csname glo@\@gls@label @index\endcsname
      \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
  \fi
}
\newcommand*{\@gls@getcounterprefix[2]}{%
  \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
    \ifx\@glo@counterprefix\@empty
      \GlossariesWarning{Hyper target '#2' can't be formed by
        prefixing^^Jlocation '#1'. You need to modify the
        definition of \string\theH\@gls@counter^^Jotherwise you
        will get the warning: "'name{\@gls@counter.#1}' has been^^J
        referenced but does not exist"}%
    \fi
  \fi
}

```



```

\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :key (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\@glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\@glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\@glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\@glsseelastsep{\space\andname\space}
\newcommand*\@glsseesep{, }

```

```

\newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*{\gls@save@numberlist}[1]{%
  \ifglssavenumberlist
    \toks@{#1}%
    \edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}\{\@do@writeaux@info}%
  \fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand{\@printglossary}[2]{%

```

```

\def\@glo@type{\glsdefaulttype}%
\def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
\def\glossarytoctitle{\glossarytitle}%
\let\org@glossarytitle\glossarytitle
\def\@glossarystyle{%
  \ifx\@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '\@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\bgroup
\@printgloss@setsort
\setkeys{printgloss}{#1}%
\@printgloss@checkexists{\@glo@type}%
{%
  \ifx\glossarytitle\org@glossarytitle
  \else
    \expandafter\let\csname @glo@type\@glo@type @title\endcsname
      \glossarytitle
  \fi
  \let\currentglossary\@glo@type
  \let\org@glossaryentrynumbers\glossaryentrynumbers
  \let\glsnonextpages\@glsnonextpages
  \let\glsnextpages\@glsnextpages
  \let\nopostdesc\@nopostdesc
  \gls@dotoc@title
  \@glossarystyle
  \let\gls@org@glossaryentryfield\glossentry
  \let\gls@org@glossarysubentryfield\subglossentry
  \renewcommand{\glossentry}[1]{%
    \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
    \gls@org@glossaryentryfield{##1}%
  }%
  \renewcommand{\subglossentry}[2]{%
    \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
    \gls@org@glossarysubentryfield{##1}{##2}%
  }%
  \@gls@preglossaryhook
  #2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}

```

```

\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglxindy
    \ifcsundef{@xdy@\@glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\@xdy@main@language}%
          }%
        }%
      }%
    }%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
              @language\endcsname}%
          }%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      }%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\@gls@codepage}%
      }%
    }%
  }%
  \@do@auxoutstuff
}
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \edef\@glo@type{#2}%

```

```

\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    {%
      }%
    }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {%
      }%
    }%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {%
      }%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{\glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{\glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{\glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{\glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{\glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{\glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{\glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{\glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@word}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=word]}{}%
}%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\cuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \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@=\uccode'#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@='#1\relax
}%
{%
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
}

```

```

    {%
      \subglossentry{\gls@level}{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \letcs{\@gls@sort}{glo\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {%
      \ifdefempty{\@gls@currentlettergroup}{}%
      {%
        \global\let\@glo@thislettergrp\@glo@thislettergrp
        \glsgroupskip
      }%
      \glsgroupheading{\@glo@thislettergrp}%
    }%
    \global\let\@gls@currentlettergroup\@glo@thislettergrp
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{%
    \def\@gls@noidxloclist@prev{%
      \forlistloop{\glsnoidxloclisthandler}{#1}%
    }
  }
  \newcommand*{\glsnoidxloclisthandler}[1]{%
    \ifdefstring{\@gls@noidxloclist@prev}{#1}%
    {%
    }%
    {%
      \@gls@noidxloclist@sep
      #1%
      \def\@gls@noidxloclist@sep{\delimN}%
      \def\@gls@noidxloclist@prev{#1}%
    }%
  }
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%

```

```

\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
\@gls@noidxloclist@sep
\@gls@noidxloclist@prev
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
\setentrycounter{#1}{#2}%
\csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
\glsdoifexistsorwarn{#2}%
{%
\ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
\ifinlistcs{#2}{\@glsref@#1}%
{}%
{\listcsgadd{\@glsref@#1}{#2}}%
\ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
{\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
{}%
\listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
}%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotoc@title\relax
}
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotoc@title\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{\@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossentrycompatibility
\csname @glsstyle@#1\endcsname}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%

```

```

\ifcase\gls@numberedsection@nr\relax
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glssubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
    \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}

```

```

}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglsentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglsentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%

```

```

\ifglssubentrycounter
  \theglossarysubentry)\space
\fi
}
\newcommand*{\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetssubentrycounter
  \fi
}
\newcommand*{\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\providecommand*{\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*{\glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glssdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}

```

```

\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\providecommand*{\compatiblesubglossentry}[3]{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@subglossentry
}
\newcommand*{\setglossentrycompatibility}{%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand{\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^^J
    I recommend you change to \string\glossentry.^^J
    If you've just upgraded, try removing your gls auxiliary
    files^^J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*{\glossarysubentryfield}[6]{%
  \GlossariesWarning

```

```

{Deprecated use of \string\glossarysubentryfield.^^J
 I recommend you change to \string\subglossentry.^^J
 If you've just upgraded, try removing your gls auxiliary
 files^^J and recompile}%
\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
 \@gls@getgrouptitle{#1}{\@gls@grptitle}%
 \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
 \dtl@ifsingle{#1}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
{%
 \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
 or test{\ifstrequal{#1}{glsnumbers}}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
{%
 \def#2{#1}%
}%
}%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
 \DTLifint{#1}%
 {\edef#2{\char#1\relax}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
}
\newcommand*{\glsgetgrouplabel}[1]{%
 \ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
 \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
}
\newcommand*{\setentrycounter}[2][1]{%
 \def\@glo@counterprefix{#1}%
 \ifx\@glo@counterprefix\@empty
 \def\@glo@counterprefix{.}%
 \else
 \def\@glo@counterprefix{.#1}%
 \fi
 \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
 \ifcsundef{@glsstyle@#1}%
 {%
 \PackageError{glossaries}{Glossary style '#1' undefined}{}%

```



```

}%
{%
  \csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^~J
      I recommend you switch to \string\setglossarystyle\space unless
      you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\csuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}

```

```

}%
{%
  \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
    \else
      \@delimR#1\delimR\delimR\%
    \fi
    \ifx\#2\%
    \else
      #2%
    \fi
    \ifx\#3\%
    \else
      \@glshypernumber#3\@nil
    \fi
  }
  \def\@delimR#1\delimR #2\delimR #3\{%
  \ifx\#2\%
    \@delimN{#1}%
  \else
    \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
  \fi}
  \def\@delimN#1{\@delimN#1\delimN \delimN\}
  \def\@delimN#1\delimN #2\delimN#3\{%
  \ifx\#3\%
    \@gls@numberlink{#1}%
  \else
    \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
  \fi
  }
  \def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil
  \endgroup}
  \def\@gls@removespaces#1 #2\@nil{%
  \toks@=\expandafter{\the\toks@#1}%
  \ifx\#2\%
    \edef\x{\the\toks@}%
    \ifx\x\empty
    \else
      \hyperlink{\glstrycounter\@glo@counterprefix\the\toks@}%
        {\the\toks@}%
    \fi
  \else
    \@gls@ReturnAfterFi{%
      \@gls@removespaces#2\@nil
    }%
  }%

```

```

\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
\newrobustcmd*\glstextup[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey{short}
\newcommand*\glsshortpluralkey{shortplural}
\newcommand*\glslongkey{long}
\newcommand*\glslongpluralkey{longplural}
\newrobustcmd*\acrfull{\@gls@hyp@opt\@ns@acrfull}
\newcommand*\@ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}{%
    {\@acrfull{#1}{#2}}[{}]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[{}]}%
}

```

```

\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
    {\@Acrfull{#1}{#2}[]}%
  }
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
    {\@ACRfull{#1}{#2}[]}%
  }
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
    {\@acrfullpl{#1}{#2}[]}%
  }
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
    {\@Acrfullpl{#1}{#2}[]}%
  }
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
    {\@ACRfullpl{#1}{#2}[]}%
  }
}

```

```

}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand*\@ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\@acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*\@newacronymhook{}
\newcommand*\@SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acrentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \gslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\@acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\@Acrfullfmt}[3]{%

```

```

\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%

```

```

\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}

```

```

}%
\renewcommand*{\genplacrfullformat}[2]{%
\glentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})%}
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}

```



```

}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\glspacespace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%

```

```

{%
  \GlsUseAcrEntryDisplayStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {%
    {\glscustomtext\glsinsert}%
  }%
}

```

```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
\glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
\Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%

```

```

{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \glshyperfirstfalse
  \renewcommand*{\genacrfullformat}[2]{%
    \protect\firstacronymfont{\gl Sentryshort{##1}}##2%
    \protect\footnote{\gl Sentrylong{##1}}%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \firstacronymfont{\Gl Sentryshort{##1}}##2%
    \protect\footnote{\gl Sentrylong{##1}}%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\gl Sentryshortpl{##1}}##2%
    \protect\footnote{\gl Sentrylongpl{##1}}%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Gl Sentryshortpl{##1}}##2%
    \protect\footnote{\gl Sentrylongpl{##1}}%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\gl Sentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\acrpluralsuffix}{\gl sacrpluralsuffix}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\gl Sentryshort{##2}}##3\space
      (\gl Sentrylong{##2})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\Gl Sentryshort{##2}}##3\space
      (\gl Sentrylong{##2})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \mfirstucMakeUppercase{\acronymfont{\gl Sentryshort{##2}}##3\space
        (\gl Sentrylong{##2})}}}%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\gl Sentryshortpl{##2}}##3\space
      (\gl Sentrylongpl{##2})}%
  }%
  \renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\Gl Sentryshortpl{##2}}##3\space
      (\gl Sentrylongpl{##2})}%
  }%
  \renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \mfirstucMakeUppercase{\acronymfont{\gl Sentryshortpl{##2}}##3\space
        (\gl Sentrylongpl{##2})}}}%
  }%
  \renewcommand*{\gl Sentryfull}[1]{%
    \acronymfont{\gl Sentryshort{##1}}\space(\gl Sentrylong{##1})}%
  \renewcommand*{\Gl Sentryfull}[1]{%
    \acronymfont{\Gl Sentryshort{##1}}\space(\gl Sentrylong{##1})}%

```

```

\renewcommand*{\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\aclong
  \let\aclp\aclongpl
  \let\Acl\Aclong
  \let\Aclp\Aclongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglssacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    }
  }

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDefaultAcronymStyle}{%
    \@for\@gls@type:=\@gls@acronymlists\do{%
        \SetDefaultAcronymDisplayStyle{\@gls@type}%
    }%
    \renewcommand{\newacronym}[4][ ]{%
        \ifx\@gls@acronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetDefaultAcronymDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \DefaultNewAcronymDef
    }%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
    \footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrno linkfootnote}[3]{%
    \footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
    \def\glsentryfmt[1]{%
        \ifdefempty\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}}%
      }%
    }%
  }%
}
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      first={\the\glsshorttok},%
      firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      text={\the\glsshorttok},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      symbol={\the\glslongtok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }

```

```

}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
      \def\@gls@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@gls@type}%
      \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
      can’t both be set}{}%
  \fi
}%
\newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\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
\ifglshassymbol{\glslabel}%
{%
\glsifplural
{%
\def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
}%
{%
\def\@glo@symbol{\glsentrysymbol{\glslabel}}%
}%
\space(\protect\firstacronymfont
{\glscapscase
{\@glo@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
}%%
}%
\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
sort={\the\glsshorttok},%
first={\the\glslongtok},%

```

```

firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDescriptionAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionNewAcronymDef
  }%
  \for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand{\acronymfont}[1]{\textsc{##1}}
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else

```

```

\ifglsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\ifglused{\glslabel}%
{%
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
{%
\firstacronymfont{\glsgenentryfmt}\gls@org@insert
\ifglshaslong{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
}%
\glsifplural
{\glentrylongpl{\glslabel}}%
{\glentrylong{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

```

\the\glskeylisttok
}%
}%
\let\@org@glsgls@assign@plural\gls@assign@plural
\let\@org@glsgls@assign@firstpl\gls@assign@firstpl
\let\@org@glsgls@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@glsgls@assign@plural
\let\gls@assign@firstpl\@org@glsgls@assign@firstpl
\let\gls@assign@descplural\@org@glsgls@assign@descplural
}
\newcommand*{\SetFootnoteAcronymStyle}{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsglsacronymlists\@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\@glsgls@type:=\@glsglsacronymlists\do{%
    \SetFootnoteAcronymDisplayStyle{\@glsgls@type}%
  }%
  \ifglsglsacrsmallcaps
    \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else
    \ifglsglsacrsmaller
      \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsglsacrdua
    \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@has@symbol{\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]{\glsgenentryfmt}%
}
\newcommand*{\DUANewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural

```

```

\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\let\gl@assign@descplural\org@gl@assign@descplural
}
\newcommand*{\SetDUASstyle}{%
  \renewcommand{\newacronym}[4][1]{%
    \ifx\@gl@acronymlists\@empty
      \def\@glo@type{acronym}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifgl@acrdescription
    \ifgl@acrfootnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifgl@acrdua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \fi

```

```

\else
  \ifglacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glacrsmallcaps}\OR
      \boolean{glacrsmaller}}{%
      {%
        \SetSmallAcronymStyle
      }%
    }%
    \ifglacrdua
      \SetDUASstyle
    \fi
  }%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}

```

```

\newcommand*{\SetCustomStyle}{%
\renewcommand{\newacronym}[4][{}]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetCustomDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\CustomNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetCustomDisplayStyle{\@gls@type}%
}%
}
\RequirePackage{glossary-hypernav}[=v4.46]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
\expandafter\show\csname glo@glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
\expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}

```

```

\newcommand*{\showgloloclist}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \csshow{glo@glstetoklabel{#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 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\@glstr@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 @useriiaaccess\endcsname{%
    \@glo@useriiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaaccess\endcsname{%
    \@glo@useriiiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter

```

```

\protected@xdef\csname glo@\glo@label @servaccess\endcsname{%
  \@glo@servaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @serviaccess\endcsname{%
  \@glo@serviaccess}%
}
\newcommand*{\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*{\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaaccess}[1]{%
  \@gls@entry@field{#1}{useriiaaccess}%
}

```

```

}
\newcommand*{\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*{\@gls@nameaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@access}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{name}{#1}{\@glsentryaccess{#2}}}%
}
\newrobustcmd*{\@gls@textaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@textaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{text}{#1}{\@glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\@gls@pluralaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@pluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{plural}{#1}{\@glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@firstaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@firstaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{first}{#1}{\@glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\@gls@firstpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\@glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@symbolaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@symbolaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\@glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\@gls@symbolpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@symbolpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\@glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@descriptionaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@descaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\@glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\@gls@descriptionpluralaccessdisplay}[2]{%

```

```

\ifcsundef{glo@glstdetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glstrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glstryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glstrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glssuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glstryuseriaccess{#2}}}%
}
\newrobustcmd*{\glssuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glstryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glssuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glssuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@userivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstryuserivaccess{#2}}}%
}
\newrobustcmd*{\glssuservaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@uservaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userv}{#1}{\glstryuservaccess{#2}}}%
}
\newrobustcmd*{\glssuserviaccessdisplay}[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}}%
            {\glssymbolpluralaccessdisplay
                {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
        }%
    }%
    {%
        \ifglsused\glslabel
        {%
            \MakeUppercase{%
                #2{\glspluralaccessdisplay
                    {\glsentryplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                        {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glssymbolpluralaccessdisplay
                        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
        {%
            \MakeUppercase{%
                #1{\glsfirstpluralaccessdisplay
                    {\glsentryfirstplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                        {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glssymbolpluralaccessdisplay
                        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
    }%
    {%
        \glscapscase
        {%
            \ifglsused\glslabel
            {%
                #2{\glstextaccessdisplay
                    {\glsentrytext{\glslabel}}{\glslabel}}%
                    {\glsdescriptionaccessdisplay
                        {\glsentrydesc{\glslabel}}{\glslabel}}%
                    {\glssymbolaccessdisplay
                        {\glsentrysymbol{\glslabel}}{\glslabel}}%
                    {\glsinsert}%
                }%
            }%
        }%
    }%

```

```

#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
  \ifglsused\glslabel
  {%
    #2{\glstextaccessdisplay
      {\Glsentrytext{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glssymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
    }%
  }%
  #1{\glsfirstaccessdisplay
    {\Glsentryfirst{\glslabel}}{\glslabel}}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glssymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \MakeUppercase{%
      #2{\glstextaccessdisplay
        {\glsentrytext{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
      }%
    }%
    \MakeUppercase{%
      #1{\glsfirstaccessdisplay
        {\glsentryfirst{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
      }%
    }%
  }%
}

```

```

        {\glsinsert}}}%
    }%
}%
}%
}%
{%
\ifglsused{\glslabel}%
{%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
{%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
}%
}
\renewcommand*{\glsгенentryfmt}{%
\ifdefempty\glscustomtext
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \ifglsused\glslabel
            {%
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
            {%
                \glsfirstpluralaccessdisplay
                {\glsentryfirstplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
        }%
        {%
            \ifglsused\glslabel
            {%
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
        }%
    }%
}

```

```

}%
{%
  \glsfirstpluralaccessdisplay
    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
  \glsinsert
}%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \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
    {%
        \ifglused\glslabel
        {%
            \glsifplural
            {%
                \glscapscase
                {%
                    \acronymfont
                    {\glsshortpluralaccessdisplay
                     {\glentryshortpl{\glslabel}}{\glslabel}}%
                    \glsinsert
                }%
            }%
            {%
                \acronymfont
                {\glsshortpluralaccessdisplay
                 {\Glsentryshortpl{\glslabel}}{\glslabel}}%
                \glsinsert
            }%
            {%
                \mfirstucMakeUppercase
                {\acronymfont
                 {\glsshortpluralaccessdisplay
                  {\glentryshortpl{\glslabel}}{\glslabel}}%
                 \glsinsert}%
            }%
        }%
    }%
    {%
        \glscapscase
        {%
            \acronymfont
            {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
            \glsinsert
        }%
    }%
}

```

```

    }%
    {%
      \acronymfont
      {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
    {%
      \mfirstucMakeUppercase
      {\acronymfont{%
        \glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
  }%
}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
  {%
    \glscapscase
    {%
      \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}%
}%
\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{\glstrytext{#2}}{#2}#3}%
}
\def\@GLstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glstextaccessdisplay{\mfirstucMakeUppercase{\glstrytext{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glstryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glstryfirst{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glstryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glstryplural{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glstryfirstplural{#2}}{#2}#3}%
}
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glstryfirstplural{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glstryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsnameaccessdisplay{\mfirstucMakeUppercase{\Glsentryname{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\Glsentrydesc{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\Glsentrydescplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#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\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#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}%
  {\glsuser@accessdisplay{\mfirstucMakeUppercase{\glsentryuser{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuser@accessdisplay{\glsentryuser{i}{#2}}{#2}#3}%
}
\def\@GLSuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuser@accessdisplay{\Glsentryuser{i}{#2}}{#2}#3}%
}
\def\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuser@accessdisplay{\mfirstucMakeUppercase{\glsentryuser{i}{#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}%
    {\glsglnameaccessdisplay{#2}{#1}}%
    {\glsgldescriptionaccessdisplay{#3}{#1}}%
    {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*{\accsuppglossarysubentryfield}[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glsglnameaccessdisplay{#3}{#2}}%
    {\glsgldescriptionaccessdisplay{#4}{#2}}%
    {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
    \ifglshaslong{\glsgllabel}{\glsglgenacfmt}{\glsglgenentryfmt}%
}%
{%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
    \renewcommand*{\genacrfullformat}[2]{%
        \glslongaccessdisplay{\glsglentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsglentryshort{##1}}}{##1})%
        }%
    \renewcommand*{\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glsglentryshort{##1}}}{##1})%
        }%
    \renewcommand*{\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsglentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsglentryshortpl{##1}}}{##1})%
        }%
    \renewcommand*{\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsglentryshortpl{##1}}}{##1})%
        }%
    \renewcommand*{\acronymentry}[1]{%
        \glsshortaccessdisplay{\acronymfont{\glsglentryshort{##1}}}{##1}%
    \renewcommand*{\acronymsort}[2]{##1}%
    \renewcommand*{\acronymfont}[1]{##1}%
    \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%

```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
          \mfirstucMakeUppercase{\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%

```



```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}})%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}})%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
      {\mfirstucMakeUppercase{\glentrylong{##2}}{##2}##3\space
      (\glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}})%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}})%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\mfirstucMakeUppercase{\glentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}})%
\renewcommand*{\glentryfull}[1]{%
  \glslongaccessdisplay{\glentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})%
}%
\renewcommand*{\glentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}%

```

```

\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \glshyperfirstfalse
  \renewcommand*{\genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
    \protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
    \protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
    \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
    \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
  }%
  \renewcommand*{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
    }
  }

```

```

        (\glslongaccessdisplay{\glsentrylong{##2}}{##2}})%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
        (\glslongaccessdisplay{\glsentrylong{##2}}{##2}})%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay
            {\mfirstucMakeUppercase
                {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
                (\glslongaccessdisplay{\glsentrylong{##2}}{##2}})}}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
            (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
            (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})%
\renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\mfirstucMakeUppercase
                {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
                (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})}}%
\renewcommand*{\glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1}})%
\renewcommand*{\Glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1}})%
\renewcommand*{\glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}})%
\renewcommand*{\Glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
        {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}})%
}
\renewacronymstyle{footnote-sc}%
{%
    \GlsUseAcrEntryDisplayStyle{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~\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter\@gobble\string\{\string" ^^J
    :close \string"\expandafter\@gobble\string\}\string" ^^J
    :attr \string"#1\string"))}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{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]{%
\ifglxsindy
  \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
  \def\@glo@range{:open-range}%
  \else
  \expandafter\if\@glo@prefix\relax
  \def\@glo@range{:close-range}%
  \fi
\fi
\protected@edef\@glslocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@glslocref
\glossary[\csname glo@#1@type\endcsname]{%
(indexentry :tkey (\csname glo@#1@index\endcsname)
:locref \string"\@glslocref\string" %
:attr \string"\@glo@suffix\string" \@glo@range
)
}%
\else
  \set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
  \glossary[\csname glo@#1@type\endcsname]{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar\@glo@numfmt}{\theglsentrycounter}}%
\fi
}
\def\@set@glo@numformat#1#2#3{%
  \expandafter\@glo@check@mkidxrangechar#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[] {#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
\ifglxsindy
  \def\writeist{%
    \openout\glswrite=\istfilename
    \write\glswrite{;; xindy style file created by the glossaries
      package in compatible-2.07 mode}%
    \write\glswrite{;; for document '\jobname' on
      \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\@xdystyle:=\@xdyrequiredstyles\do{%
      \ifx\@xdystyle\@empty

```

```

\else
  \protected@write\glswrite{}\{(require
    \string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-lowercase\string")
    :min-range-length \@glsminrange)}%
\else
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (:sep "\@gls@roman")
    :min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
  \string"Roman-page-numbers\string"^^J\space\space\space
  (\string"roman-numbers-uppercase\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-page-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"alpha-page-numbers\string"^^J\space\space\space
  (\string"alpha\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
    :sep \string"\glscpositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmrange)}}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glssseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
    \glossarysection[\string\glossarytoctitle]{\string
    \glossarytitle}\string\glossarypreamble\string~n\string\begin
    {theglossary}\string\glossaryheader\string~n\string" ^^J\space
    \space\space:close \string"\expandafter\@gobble
    \string~n\string~n\string
    \end{theglossary}\string\glossarypostamble
    \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glresetentrylist
    \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsoopenbrace\string\glossaryentrynumbers
    \glsoopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi

```

```

\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string\string\glsgroupskip\string~n\string)}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string\string\glsgroupheading
  \glsopenbrace\string^^J\space\space\space
  :close-head \string\glsclosebrace\string)}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
  \openout\glswrite=\istfilename
  \write\glswrite{\expandafter\@gobble\string\% makeindex style file
    created by the glossaries package}
  \write\glswrite{\expandafter\@gobble\string\% for document
    'jobname' on \the\year-\the\month-\the\day}
  \write\glswrite{actual '\@gls@actualchar'}
  \write\glswrite{encap '\@gls@encapchar'}
  \write\glswrite{level '\@gls@levelchar'}
  \write\glswrite{quote '\@gls@quotechar'}
  \write\glswrite{keyword \string\string\glossaryentry\string"}
  \write\glswrite{preamble \string\string\glossarysection[\string
    \glossarytoctitle]{\string\glossarytitle}\string
    \glossarypreamble\string~n\string\begin{theglossary}\string
    \glossaryheader\string~n\string"}
  \write\glswrite{postamble \string\string\string~n\string
    \end{theglossary}\string\glossarypostamble\string~n
    \string"}
  \write\glswrite{group_skip \string\string\glsgroupskip\string~n
    \string"}
  \write\glswrite{item_0 \string\string\string~n\string"}
  \write\glswrite{item_1 \string\string\string~n\string"}
  \write\glswrite{item_2 \string\string\string~n\string"}
  \write\glswrite{item_01 \string\string\string~n\string"}
  \write\glswrite{item_x1

```

```

\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string\}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
\string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
\string"\string\}\string\relax
\string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glssymbols\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
\ifcsundef{@glscompstyle@#1}%
{%
\csdef{@glscompstyle@#1}{#2}%
}%
}

```



```

    {%
      \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
    }%
  }
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
        {%
          \glsresetsubentrycounter
          \glsinlineparentchildseparator
          \def\gls@inlinesubsep{}%
          \def\gls@inlinepostchild{\glsinlinepostchild}%
        }%
      {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
\renewcommand{\glossarysubentryfield}[6]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{##3}%
  \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
}
\compatglossarystyle{list}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
  }
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%

```

```

\renewcommand*{\glossaryentryfield}[5]{%
  \item[\glentryitem{##1}\glstarget{##1}{##2}]%
  \mbox{}\par\nobreak\@afterheading
  ##3\glspostdescription\space ##5}%
\renewcommand{\glossarysubentryfield}[6]{%
  \par
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
  }%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
  }%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%

```

```

\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
    \tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
    &

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
        \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscmpstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscmpstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscmpstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%

```

```

\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%

```

```

\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{alttree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else
\settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[Opt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription \space ##5\par
\def\@gls@prevlevel{0}%

```

```

}%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \ifundefined{\@glswidestname\romannumeral##1}{%
      \settowidth{\@gls@tmplen}{\textbf{\@glswidestname\space}}{%
      \settowidth{\@gls@tmplen}{\textbf{%
        \csname \@glswidestname\romannumeral##1\endcsname\space}}}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\@gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \ifundefined{\@glswidestname\romannumeral\@gls@prevlevel}{%
        \settowidth{\glstreeindent}{\textbf{%
          \@glswidestname\space}}}%
        \settowidth{\glstreeindent}{\textbf{%
          \csname \@glswidestname\romannumeral\@gls@prevlevel
            \endcsname\space}}}%
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
      \fi
    \fi
    \hangindent\glstreeindent
    \makebox[0pt][r]{\makebox[\@gls@tmplen][l]{%
      \textbf{\glstarget{##2}{##3}}}%
    \ifx##5\relax\relax
    \else
      (##5)\space
    \fi
    ##4\glspostdescription\space ##6\par
    \def\@gls@prevlevel{##1}%
  }%
}%
\compatglossarystyle{almtreegroup}{%
  \csuse{\@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
  \csuse{\@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
  \csuse{\@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
  \csuse{\@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%

```

```

\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenoname}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalttree}{%
\csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{mcolalttreegroup}{%
\csuse{@glscompstyle@alttree}%
}%
\compatglossarystyle{mcolalttreehypergroup}{%
\csuse{@glscompstyle@alttree}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%

```



```

\renewcommand*{\glossaryentryfield}[5]{%
  \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%

```

```

\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%

```

8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \PackageWarningNoLine{glossaries}%
      {No language module detected for ‘\this@dialect’.\MessageBreak
      Language modules need to be installed separately.\MessageBreak
      Please check on CTAN for a bundle called\MessageBreak
      ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
{%
}
```

8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2020/03/19 v4.46 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[=v4.46]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
```

```

\let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label\{prefix\}\@glo@entryprefix}%
\gls@assign@field{}\@glo@label\{prefixplural\}\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
\{csname glo@\@glo@label @prefix\endcsname\}\@glo@label\{prefixfirst}%
\{glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
\{csname glo@\@glo@label @prefixplural\endcsname\}\@glo@label}%
\{prefixfirstplural\}\@glo@entryprefixfirstplural}%
}
\newcommand*\@glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*\@glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*\@glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*\@glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*\@Glsentryprefixfirst}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\@ifglshasprefix}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixplural}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirst}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixfirst}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirstplural}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixfirstplural}%

```

```

    {#3}%
    {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
    {\@glsxtr@record{#1}{#2}{glslink}}%
  }%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
    \@gls@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirstplural{#2}%
      {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}

```

```

\newrobustcmd{\Pgl}{\@gls@hyp@opt\@Pgl}
\newcommand*{\@Pgl}[2][{}]{%
  \new@ifnextchar[%
    {\@Pgl@{#1}{#2}}%
    {\@Pgl@{#1}{#2}[]}%
}
\def\@Pgl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}%
      {%
        \Glsentryprefix{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {%
        \Glsentryprefixfirst{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
  }%
}
\newrobustcmd{\Pglsp}{\@gls@hyp@opt\@Pglsp}
\newcommand*{\@Pglsp}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglsp@{#1}{#2}}%
    {\@Pglsp@{#1}{#2}[]}%
}
\def\@Pglsp@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glsentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}

```

```

}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@glshyp@opt\PGLS}
\newcommand*{\@PGLS}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2}[]}%
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\PGLSpl}{\@glshyp@opt\PGLSpl}
\newcommand*{\@PGLSpl}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLSpl@{#1}{#2}}%
    {\@PGLSpl@{#1}{#2}[]}%
}
\def\@PGLSpl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%

```

```

\ifglshasprefixfirstplural{#2}%
{\mfirstucMakeUppercase{\glstryprefixfirstplural{#2}\glsprefixsep}}{}%
}%
\@GLSpl@{#1}{#2}{#3}%
}%
}

```

8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2020/03/19 v4.46 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \edef\gls@grplabel{#2}\protected@edef\@gls@grptitle{#3}%
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’~Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    \fi
  }
  \let\gls@hypergroup@prerun\relax
  \AtEndDocument{\gls@hypergroup@prerun}
  \newcommand*{\@gls@hypergroup}[2]{%
    \ifundefined{\@gls@hypergroup@list@#1}{%
      \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
    }{%
      \expandafter\let\expandafter\@gls@tmp
        \csname @gls@hypergroup@list@#1\endcsname
      \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
        \@gls@tmp,#2}%
    }%
  }
  }
  \newcommand*{\glsnavigation}{%
    \def\@gls@between{}%
    \ifcsundef{\@gls@hypergroup@list@\@glo@type}%

```



```

{%
  \def\@gls@list{%
}%
{%
  \expandafter\let\expandafter\@gls@list
    \csname @gls@hypergroup@list@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
  \@gls@between
  \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
  \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
  \let\@gls@between\glshypernavsep
}%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glssymbolnav}{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}

```

8.9 Rollback v4.46 (glossary-inline-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2020/03/19 v4.46 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{%
    \def\gls@inlinesubsep{%
    \def\gls@inlinepostchild{%
    }%
    {\glspostinline}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \glsentryitem{##1}%
    \glsinlinenameformat{##1}%
    \glossentryname{##1}%
  }%
  \ifglsdescsuppressed{##1}%
  {%
    \glsinlineemptydescformat
    {%
      \glossentrysymbol{##1}%
    }%
  }%
}

```

```

    {%
      ##2%
    }%
  }%
  {%
    \ifglshasdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglshaschildren{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{%
  \newcommand*{\glspostinline}{\glspostdescription\space}
  \newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
  \newcommand*{\glsinlinedescformat}[3]{\space#1}
  \newcommand*{\glsinlineemptydescformat}[2]{%
  \newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
  \newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.10 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%

```

```

\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt[1]{#1}
\newcommand*\glslistnavigationitem[1]{\item[#1]}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\begin{description}}{\end{description}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading[1]{}%
  \renewcommand*\glossentry[2]{%
    \item[\glstentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glslnavigation}}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt
      {\glslnavhypertarget{##1}{\glsgrouptitle{##1}}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry[2]{%
    \item[\glstentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glslnavigation}}%

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
  \renewcommand*{\subglossentry}[3]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
  }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
}

```

8.11 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{%
}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{%
}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{\glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand*{\subglossentry}[3]{%
    &

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
    \fi
}
\newglossarystyle{longborder}{%
    \setglossarystyle{long}%
    \renewenvironment{theglossary}{%
        \begin{longtable}{|l|p{\glstdescwidth}|}{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longheader}{%
    \setglossarystyle{long}%
    \renewcommand*{\glossaryheader}{%
        \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
\newglossarystyle{longheaderborder}{%
    \setglossarystyle{longborder}%
    \renewcommand*{\glossaryheader}{%
        \hline\bfseries \entryname & \bfseries
        \descriptionname\tabularnewline\hline
        \endhead
        \hline\endfoot}%
}
\newglossarystyle{long3col}{%
    \renewenvironment{theglossary}{%
        \begin{longtable}{lp{\glstdescwidth}p{\glspagelistwidth}}%
        {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupeheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
    \fi

```

```

}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glstdescwidth}|p{\glspagelistwidth}|}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&

```

```

        \bfseries \symbolname&
        \bfseries\pagelistname\tabularnewline\endhead}%
    }
    \newglossarystyle{long4colborder}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|l|l|l|l|}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
    }
    \newglossarystyle{long4colheaderborder}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|l|l|l|l|}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{%
            \hline\bfseries\entryname&\bfseries\descriptionname&
            \bfseries \symbolname&
            \bfseries\pagelistname\tabularnewline\hline\endhead
            \hline\endfoot}%
    }
    \newglossarystyle{altlong4col}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colheader}{%
        \setglossarystyle{long4colheader}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colborder}{%
        \setglossarystyle{long4colborder}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colheaderborder}{%
        \setglossarystyle{long4colheaderborder}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
            {\end{longtable}}%
    }
}

```

8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput

```



```

\setglossarystyle{long4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@ \pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
          \fi
          \outputpage
        \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
      \fi
    }%
  }%
}

```

```

\fi
\fi
\global\@colroom\@colht
\global\vsizel\@colht
{\unvbox\z@box\ifvoidLT@lastfootLT@footelseLT@lastfoot\fi}%
\fi
\else
\setbox\@cclv\box{\unvbox\@cclv\copyLT@foot\vss}%
\@makecol
\@outputpage
\global\vsizel\@colroom
\copyLT@head
\glstpenaltycheck
\nobreak
\fi
}%
}

```

8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
\newlength{glsdescwidth}
\setlength{glsdescwidth}{0.6\hsizel}
}{%}
\@ifundefined{glspagelistwidth}{%
\newlength{glspagelistwidth}
\setlength{glspagelistwidth}{0.1\hsizel}
}{%}
\newglossarystyle{longragged}{%
\renewenvironment{theglossary}{%
{\begin{longtable}{l>{\raggedright}p{glsdescwidth}}}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1}\glspostdescription\space ##2%
\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2}%
\glspostdescription\space ##3%
\tabularnewline
}

```

```

}%
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}||}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
  }
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2} &
      ##3\tabularnewline
    }%
  }
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|}%
      >{\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{\glsgdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \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{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

        {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
          >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}}%
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.46]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }
  }

```

```

    }%
    {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcoltree}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}

```

```

\renewcommand*{\glsgroupeheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
\renewcommand*{\glsgroupeheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand{\glsgroupeheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupeheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%

```

```

}%
{\end{multicols}}}%
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
  \indexspace}%
}
\newglossarystyle{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalttreegroup}{%
  \setglossarystyle{mcolalttree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalttreehypergroup}{%
  \setglossarystyle{mcolalttree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%

```



```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

8.15 Rollback v4.46 (glossary-super-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%

```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheader}{%
        \setglossarystyle{super}%
        \renewenvironment{theglossary}%
        {\tablehead{\bfseries \entryname &
            \bfseries\descriptionname\tabularnewline}%
            \tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}}}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheaderborder}{%
        \setglossarystyle{super}%
        \renewenvironment{theglossary}%
        {\tablehead{\hline\bfseries \entryname &
            \bfseries \descriptionname\tabularnewline\hline}%
            \tabletail{\hline}}
        {\begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{super3col}{%
        \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
        \renewcommand*{\glossaryheader}{}%
        \renewcommand*{\glsgroupheading}[1]{}%
        \renewcommand{\glossentry}[2]{%
            \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
            \glossentrydesc{##1} & ##2\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
            &
            \glssubentryitem{##2}%
            \glstarget{##2}{\strut}\glossentrydesc{##2} &
            ##3\tabularnewline
        }%
        \ifglsgroupskip
            \renewcommand*{\glsgroupskip}{}%
        \else
            \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
        \fi
    }
    \newglossarystyle{super3colborder}{%
        \setglossarystyle{super3col}%
        \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
            \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
            {\end{supertabular}}%
    }

```

```

}
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsglwidth}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{\glsglwidth}|p{\glspagelistwidth}|}%
      {\end{supertabular}}}%
    }%
}
\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}%
    {\begin{supertabular}{llll}}{%
      \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}{llll}}%
        {\end{supertabular}}%
    }
    \newglossarystyle{super4colborder}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline}\tabletail{\hline}%
             \begin{supertabular}{|l|l|l|l|}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{super4colheaderborder}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
                        \bfseries\symbolname &
                        \bfseries\pagelistname\tabularnewline\hline}%
             \tabletail{\hline}%
             \begin{supertabular}{|l|l|l|l|l|}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4col}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{} \tabletail{}}%
            {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colheader}{%
        \setglossarystyle{super4colheader}%
        \renewenvironment{theglossary}%
            {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
                        \bfseries\symbolname &
                        \bfseries\pagelistname\tabularnewline}\tabletail{}}%
            {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colborder}{%
        \setglossarystyle{super4colborder}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline}\tabletail{\hline}%
             \begin{supertabular}%
                 {l|lp{\glsgdescwidth}|l|lp{\glspagelistwidth}|}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colheaderborder}{%
        \setglossarystyle{super4colheaderborder}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline
                        \bfseries\entryname &
                        \bfseries\descriptionname &

```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
        {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}%
    \end{supertabular}}%
}

```

8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
    \newlength{glsdescwidth}
    \setlength{glsdescwidth}{0.6\hsize}
}{%
}
\@ifundefined{glspagelistwidth}{%
    \newlength{glspagelistwidth}
    \setlength{glspagelistwidth}{0.1\hsize}
}{%
}
\newglossarystyle{superragged}{%
    \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}}%
        {\begin{supertabular}{1>\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\space ##2%
        \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3%
        \tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}
\newglossarystyle{superraggedborder}{%
    \setglossarystyle{superragged}%
}

```

```

\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
   \tabularnewline}%
   \tabletail{}}%
  {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
   \bfseries \descriptionname\tabularnewline\hline}%
   \tabletail{\hline}%
   \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
  {\end{supertabular}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
  {\tablehead{} \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{\glstdescwidth}|%
>{\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{\glstdescwidth}%
>{\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{\glstdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
{\end{supertabular}}%
}

\newglossarystyle{altsuperragged4col}{%
\renewenvironment{theglossary}%
{
\tablehead{}\tabletail{}%
\begin{supertabular}{|l>{\raggedright}p{\glstdescwidth}l%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupeheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{altsuperragged4colheader}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsgdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
  \setglossarystyle{altsuper4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {ll>{\raggedright}p{\glsgdescwidth}ll}%
      >{\raggedright}p{\glspagelistwidth}ll}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {ll>{\raggedright}p{\glsgdescwidth}ll}%
      >{\raggedright}p{\glspagelistwidth}ll}%
    {\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}\glstreenamfmt{\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
    \glstreenamfmt{\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}{\glsgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{tree}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
\setglossarystyle{tree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
\indexspace}%
}
\newglossarystyle{treehypergroup}{%
\setglossarystyle{tree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%

```

```

\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt
    {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglshasparent{\@glo@label}%
      {}%
    }%
  }%
}

```

```

\settowidth{\dimen@}{%
  {\glstreenamefmt{\glstentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
  \gls@tmplen=\dimen@
  \letcs{\@glswidestname}{glo@glstdetoklabel{\@glo@label}@name}%
\fi
}%
}%
}%
}
\newcommand*{\glssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
    #2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
\newglossarystyle{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[0pt][r]{\glstreenamebox{\glstreeindent}{%
      \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
    \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
    \glossentrydesc{##1}\glspostdescription \space ##2\par
    \def\@gls@prevlevel{0}%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
      \@ifundefined{@glswidestname\romannumeral##1}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname}\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength{\glstreeindent}{\gls@tmplen

```

```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}}%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 Rollback v4.49 (glossaries-2021-11-01.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareCurrentRelease{v4.49}{2021-11-01}
\ProvidesPackage{glossaries}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
{%
  \@gls@docloadedtrue
}%
}%
{%
  \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\@gls@debug@val\@gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\@gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*{\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*{\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \let\@glsshowtarget\@gobble
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
  \or

```

```

% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget[1]{%
\ifmmode
  \nfss@text{\glsshowtargetfont [#1]}%
\else
  \ifinner
    {\glsshowtargetfont [#1]}%
  \else
    \glsshowtargetouter{#1}%
  \fi
\fi
}
\newcommand*\glsshowtargetouter[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget[1]{%
\newrobustcmd*\@@glsshowtarget[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%

```

```

\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
  subsection,subsubsection,paragraph,subparagraph}[section]{%

```



```

\renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}%
\newcommand*{\@glossaryseclabel}{}%
\newcommand*{\glsautoprefix}{%
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{%
\label{\glsautoprefix@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
\renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{%
\@gls@declareoption{seeautonumberlist}{%
\renewcommand*\@glo@seeautonumberlist{%
\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationstrue
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}[v4.49]}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}}

```

```

\IfFileExists{supertabular.sty}{%
  \newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}[=v4.49]}{%
    \newcommand*{\@gls@loadsuper}{}}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}}
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}[=v4.49]}
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
    \ifdefstring{\@glossary@default@style}{list}%
    {\let\@glossary@default@style\relax}%
    }%
  }%
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}[=v4.49]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
    \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  {%
  }%
}
\fi
}

```

```

\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
}%
}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \def\@gls@defsort##1##2{%
      \ifx\@glo@sort\@gls@defaultsort
        \let\@glo@sort\@glo@name
      \fi
      \let\glsdosanitizesort\@gls@sanitizesort
      \glsprestandardsort{\@glo@sort}{##1}{##2}%
      \expandafter\protected@xdef\csname glo###2@sort\endcsname{\@glo@sort}%
    }%
  \def\@gls@setsort##1{%
    \let\@glo@check@sortallowed\@gobble
  }
}
\@gls@setupsort@standard
\newcommand*\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*{\@glsacronymlists}{%
\newcommand*{\@addtoacronymlists}[1]{%
    \ifx\@glsacronymlists\@empty
        \protected@xdef\@glsacronymlists{#1}%
    \else
        \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
    \fi
}
\newcommand*{\DeclareAcronymList}[1]{%
    \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
    \protected@edef\@do@gls@islistofacronyms{%
        \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
    \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
    \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
        \def\@gls@before{##1}\def\@gls@after{##2}}%
    \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
    \ifx\@gls@after\@nnil
        #4%
    \else
        #3%
    \fi
}
\newif\if@glsisacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
    \glsIfListOfAcronyms{#1}%
    {\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
    \renewcommand*{\@glsacronymlists}{#1}%
}

```

```

\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
}

```



```

\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefined\@glo@sort
{}%
{%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent\@secondoftwo
\let\@text@composite\@secondoftwo
\let\@tabacckludge\@secondoftwo
\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let\'\'@firstofone
\let\'\'@firstofone
\let\~\@firstofone
\let\"\"@firstofone
\let\u\@firstofone
\let\t\@firstofone
\let\d\@firstofone
\let\r\@firstofone
\let\=\@firstofone
\let\.\@firstofone
\let\~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%

```

```

\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glSsetnoexpandfield{desc}%
\glSsetnoexpandfield{descplural}%
\else
\glSsetexpandfield{desc}%
\glSsetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glSsetnoexpandfield{name}%
\else
\glSsetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glSsetnoexpandfield{symbol}%
\glSsetnoexpandfield{symbolplural}%
\else
\glSsetexpandfield{symbol}%
\glSsetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
\ifglssanitizesort
\glSsetnoexpandfield{sortvalue}%
\renewcommand*{\@glS@noidx@setsanitizesort}{%
\glssanitizesorttrue
\glSsetnoexpandfield{sortvalue}%
}%
\else
\glSsetexpandfield{sortvalue}%

```

```

\renewcommand*\@gls@noidx@setsanitizesort}{%
  \glssanitizesortfalse
  \glssetexpandfield{sortvalue}%
}%
\fi
}
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
\newcommand*\@gls@noidx@setsanitizesort}{%
  \glssanitizesortfalse
  \glssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}[true,false][true]{%
  \setbool{glssanitizesort}{#1}%
  \ifglssanitizesort
    \glssetnoexpandfield{sortvalue}%
  \else
    \glssetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glssetexpandfield{name}%
      \glssetexpandfield{symbol}%
      \glssetexpandfield{symbolplural}%
      \glssetexpandfield{desc}%
      \glssetexpandfield{descplural}%
    }%
  }%
  \setkeys[gls]{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
}

```

```

        {}%
    }%
    {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
    \glsifusetranslator
    {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
    {#3}%
}
\@gls@declareoption{nottranslate}{%
    \glstranslatefalse
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
    [\gls@translate@val\gls@translate@nr]%
    {true,false,babel}[true]%
    {%
        \ifcase\gls@translate@nr\relax
            \glstranslatetrue
            \renewcommand*\@gls@usetranslator{%
                \@ifpackageloaded{polyglossia}%
                {%
                    \let\glsifusetranslator\@secondoftwo
                }%
                {%
                    \@ifpackageloaded{babel}%
                    {%
                        \IfFileExists{translator.sty}%
                        {%
                            \RequirePackage{translator}%
                            \let\glsifusetranslator\@firstoftwo
                        }%
                    }%
                }%
            }%
        }%
    }%
}
\or
    \glstranslatefalse
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
\or
    \glstranslatetrue
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse

```

```

\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
  \glstranslatetrue
  \let\glsifusetranslator\@firstoftwo
}%
{%
  \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
  {
    \@ifpackageloaded{\gls@thissty}%
    {%
      \glstranslatetrue
      \@endfortrue
    }%
    {}%
  }
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{%
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{%
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{%
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifbool{glsacrdescription}%
{}%
}%
\renewcommand*{\@gls@sanitizedesc}{}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{%
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{%
\define@choicekey{glossaries.sty}{order}{word,letter}{%

```

```

\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{%
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\language}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}

```

```

}%
{%
  \ifdefequal\@makeglossary\@gobble
  {%
    \GlossariesWarning{Option 'restoremakegloss' has no effect
      (\string\makeglossaries\space has already been used)}%
  }%
  {%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been restored}%
    \let\@domakeglossaries\@firstofone
  }%
}%
}
\newcommand*{\@do@write@glslabels}{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
  \newwrite\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@glsl@fulllist{\@ignored@glossaries}}%
  \forallglseentries[\@glsl@fulllist]{\@glseentry}%
  {%
    \ifdefempty{\@glseentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glsl@thisentryname{%
      \else
        \letcs{\@glsl@thisentryname}{glo\@glsdetoklabel{\@glseentry}@name}%
        \@onelevel@sanitize\@glsl@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@glseentry\do@write@glslabelsname{\@glsl@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*{\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglslautomake

```

```

\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
      \PackageError{glossaries}{You must use
        \string\makeglossaries\space with automake=true}
        {%
          Either remove the automake=true setting or
          add \string\makeglossaries\space to your document preamble.%
        }%
    }%
  \else
    \renewcommand*{\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{%}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%}
\ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
  {%}
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
    glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
    instead. If you \MessageBreak need the original definition of
    \string\makeglossary\space use \MessageBreak the package options

```



```

kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict))}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
\GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict))}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict))}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%

```

```

\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]]{%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{}
\newcommand*{\@gls@numbersdef}{%
\newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
\newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
\newcommand*{\gls@tr@set@numbers@toctitle}{%
\translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
}%
}%
\@gls@declareoption{index}{%
\ifx\@gls@do@indexdef\@empty
\let\@gls@do@indexdef\@gls@indexdef
\fi
}
\@gls@declareoption{noglossaryindex}{%
\let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{}
\newcommand*{\@gls@indexdef}{%
\newglossary[ilg]{index}{ind}{idx}{\indexname}%
\newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
\newcommand*{\newterm}[2][\%
\newglossaryentry{##2}%
{type={index},name={##2},description={\nopostdesc},##1}}
\let\@gls@do@indexdef\relax
}%
}%
\@for\CurrentOption :=\@declaredoptions\do{%
\ifx\CurrentOption\@empty
\else
\@expandtwoargs
\in@ {\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
\ifin@
\@use@option
\expandafter \let\csname ds@\CurrentOption\endcsname\@empty
\fi
\fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.49]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,nottranslate,nolong,nosuper,notree,nostyles,%

```

```

    nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglssubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
  {%
    \ifcsundef{chapter}{}%
    {%
      \let\@gls@old@chapter\@chapter
      \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
      \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}}%
    }%
  }%
  {}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
  \ifx\@gls@onlypremakeg\@empty

```

```

\def\@gls@onlypremakeg{#1}%
\else
\expandafter\toks@\expandafter{\@gls@onlypremakeg}%
\edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
\fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
\def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}
\providecommand*{\glsymbolsgroupname}{Symbols}
\providecommand*{\glsnumbersgroupname}{Numbers}
\newcommand*{\glspluralsuffix}{s}
\newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*{\glsupacrpluralsuffix}{\glsup{t}\glsacrpluralsuffix}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
\ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
\ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*{\addglossarytocaptions}[1]{}
\ifglstranslate
\RequirePackage{tracklang}
\@gls@usetranslator
\ifpackageloaded{translator}
{%
\ifboolexpr
{
test {\ifdefstring{\trans@languages}{English}}
and not
test {\ifdefstring{bbl@loaded}{english}}
}
{%
\let\glsifusetranslator\@secondoftwo

```

```

}%
{%
  \usedictionary{glossaries-dictionary}%
  \renewcommand*{\addglossarytocaptions}[1]{%
    \ifcsundef{captions#1}{}%
    {%
      \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
      \expandafter\toks@\expandafter{\@gls@tmp
      \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
{}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%

```

```

        \renewcommand*{\glssymbolsgroupname}{%
            \translate{Symbols (glossaries)}}%
        \renewcommand*{\glsnumbersgroupname}{%
            \translate{Numbers (glossaries)}}%
    }{}%
\fi
\DeclareRobustCommand*{\nopostdesc}{}
\newcommand*{\@nopostdesc}{%
    \let\org@glspostdescription\glspostdescription
    \def\glspostdescription{%
        \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
    \renewcommand*{\gls@istfilebase}{#1}%
    \ifglsxindy
        \def\istfilename{\gls@istfilebase.xdy}
    \else
        \def\istfilename{\gls@istfilebase.ist}
    \fi
}
\@onlypremakeg\setStyleFile
\ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
\else
    \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*{\gls@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{}
\newcommand*{\glscompositor}{.}
\newcommand*{\glsSetCompositor}[1]{%
    \renewcommand*{\glscompositor}{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*{\@glsAlphacompositor}{\glscompositor}
\ifglsxindy
    \newcommand*{\glsSetAlphaCompositor}[1]{%
        \renewcommand*{\@glsAlphacompositor}{#1}}
\else
    \newcommand*{\glsSetAlphaCompositor}[1]{%
        \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{}
\newcommand*{\glsSetSuffixF}[1]{%
    \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{}
\newcommand*{\glsSetSuffixFF}[1]{%
    \renewcommand*{\gls@suffixFF}{#1}}

```

```

}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*{\glsnumberformat}[1]{\glsnumber{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUChead{#1}}{\memUChead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else

```

```

        \@mkboth{#1}{#1}%
    \fi
}
}
\providecommand{\glossarymark}[1]{%
    \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
    \else
        \@mkboth{#1}{#1}%
    \fi
}
\newcommand*{\setglossarysection}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossarysection}[2]{%
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname[#1]{#2}%
    }%
    {%
        \csname\@glossarysec\endcsname*{#2}%
        \@gls@toc{#1}{\@glossarysec}%
    }%
    \@glossaryseclabel
}
\newcommand*{\@p@glossarysection}[2]{%
    \glsclearpage
    \phantomsection
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}{\@glossarysec}%
        \csname\@glossarysec\endcsname*{#2}%
    }%
    \@glossaryseclabel
}
\newcommand*{\gls@docclearpage}{%
    \ifthenelse{\equal{\@glossarysec}{chapter}}{%
    {%
        \ifcsundef{cleardoublepage}%
        {%
            \clearpage
        }%
        {%
            \ifcsdef{if@openright}%
            {%
                \if@openright
                \cleardoublepage
            }%
        }%
    }%
    }%
}

```



```

\else
\clearpage
\fi
}%
{%
\cleardoublepage
}%
}%
}%
{%
}%
}
\newcommand*{\glsclearpage}{\gls@docclearpage}
\newcommand*{\@gls@toc}[2]{%
\ifglstoc
\ifglsnumberline
\addcontentsline{toc}{#2}{\protect\numberline{#1}}%
\else
\addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*{\glsnoindywarning}[1]{%
\GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*{\glsnomakeindexwarning}[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
\edef\xdyattributelist{%
\fi
\ifglsxindy
\def\xdylocref{}
\fi
\newcommand*{\@gls@ifinlist}[4]{%
\def\@do@ifinlist##1,#1,##2\end@doifinlist{%
\def\@gls@listsuffix{##2}%
\ifx\@gls@listsuffix\@empty
#4%
\else
#3%
\fi
}%
\@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
\newcommand*{\@xdycounters}{\glscounter}
\newcommand*{\GlsAddXdyCounters}[1]{%

```

```

\@for\@gls@ctr:=#1\do{%
  \edef\@do@addcounter{%
    \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
    {%
      \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
        \noexpand\@gls@ctr}%
    }%
  }%
  \do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
\newcommand*\@glsaddxdyattribute[2]{%
  \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
    \string"#2#1\string"%
  \expandafter\toks@\expandafter{\@xdylocref}%
  \edef\@xdylocref{\the\toks@ ^^J%
    (markup-locref
  :open \string"\glstildechar n%
    \expandafter\string\csname glsX#2X#1\endcsname
    \string" ^^J
  :close \string"\string" ^^J
  :attr \string"#2#1\string")}%
  \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
    \setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
  }%
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
  \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \@for\@this@counter:=\@xdycounters\do{%
    \protected@edef\gls@do@addxdyattribute{%
      \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
    }
    \gls@do@addxdyattribute

```

```

    }%
    \let\GlsAddXdyCounters\@disabled@gl saddx dy counters
  }
  \@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \gl sn ox indy warning\GlsAddXdyAttribute}
\fi
\if gl sx indy
\newcommand*{\@ gl s@ add predefined attributes}{%
  \GlsAddXdyAttribute{gl s number format}
  \GlsAddXdyAttribute{text rm}
  \GlsAddXdyAttribute{text sf}
  \GlsAddXdyAttribute{text tt}
  \GlsAddXdyAttribute{text bf}
  \GlsAddXdyAttribute{text md}
  \GlsAddXdyAttribute{text it}
  \GlsAddXdyAttribute{text up}
  \GlsAddXdyAttribute{text sl}
  \GlsAddXdyAttribute{text sc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{gl s hyper number}
  \GlsAddXdyAttribute{hyper rm}
  \GlsAddXdyAttribute{hyper sf}
  \GlsAddXdyAttribute{hyper tt}
  \GlsAddXdyAttribute{hyper bf}
  \GlsAddXdyAttribute{hyper md}
  \GlsAddXdyAttribute{hyper it}
  \GlsAddXdyAttribute{hyper up}
  \GlsAddXdyAttribute{hyper sl}
  \GlsAddXdyAttribute{hyper sc}
  \GlsAddXdyAttribute{hyper emph}
  \GlsAddXdyAttribute{gl s ignore}
}
\else
  \let\@ gl s@ add predefined attributes\relax
\fi
\def\@ x dy user alphabets{}
\if gl sx indy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@ x dy user alphabets{%
      \@ x dy user alphabets ~^J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \gl sn ox indy warning\GlsAddXdyAlphabet}
\fi
\if gl sx indy
  \newcommand*{\@ gl s@ x dy@ location list}{%
    roman-page-numbers,%

```

```

    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
    \expandafter
        \edef\csname @gls@xdy\Lclass@roman-page-numbers\endcsname{%
            \string"roman-numbers-lowercase\string"%
        }%
\else
    \expandafter
        \edef\csname @gls@xdy\Lclass@roman-page-numbers\endcsname{%
            :sep \string"\@gls@roman\string"%
        }%
\fi
\expandafter\def\csname @gls@xdy\Lclass@Roman-page-numbers\endcsname{%
    \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@arabic-page-numbers\endcsname{%
    \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@alpha-page-numbers\endcsname{%
    \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@Alpha-page-numbers\endcsname{%
    \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@Appendix-page-numbers\endcsname{%
    \string"ALPHA\string"
    :sep \string"\@gls@alphacompositor\string"
    \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy\Lclass@arabic-section-numbers\endcsname{%
    \string"arabic-numbers\string"
    :sep \string"\glscompositor\string"
    \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxsindy

```

```

\newcommand\@xdycrossrefhook{}
\fi
\ifglxsindy
\newcommand*\GlsAddXdyLocation}[3][{}]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glssopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glssopenbrace"
#1
:sep "\glsclosebrace\glssopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnosindywarning\GlsAddXdyLocation}
\fi
\ifglxsindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxsindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnosindywarning\GlsSetXdyLocationClassOrder}

```

```

\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))}%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnxindywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnxindywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnxindywarning\GlsSetXdyStyles}
\fi
\newcommand*\@findrootlanguage{-}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glstypetype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnxindywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\@GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }

```

```

\AtBeginDocument{%
  \ifx\gls@codepage\empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglxindy@glxnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glxnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{\fi}%
  }%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forallglsentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand*\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

```

```

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%

```



```

\protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\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}%
  {%
    \ifcsempy{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsempy{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
  {#2}%
  {#3}%
}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

```

```

    }%
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo\glsdetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
      {#3}%
    {#2}%
  }%
}
}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo\glsdetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        {%
          \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
              #4%
            }%
          {%
            \let\glscurrentfieldvalue\@glo@thisvalue
            #3%
          }%
        }%
      }%
    }%
    {%
      \@gls@fetchfield{\@gls@thisfield}{#1}%
      \ifdef\@gls@thisfield
        {%
          \letcs{\@glo@thisvalue}{glo\glsdetoklabel{#2}@#1}%
          \ifdefempty\@glo@thisvalue
            {%
              #4%
            }%
          {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
              {%
                #4%
              }%
            {%
              \let\glscurrentfieldvalue\@glo@thisvalue
              #3%
            }%
          }%
        }%
      }%
    }%
  }%

```

```

        }%
    }%
    }%
    {%
        \GlossariesWarning{Unknown entry field ‘#1’}%
        #4%
    }%
    }%
    }%
}
\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{@glo@types}{{},}
\newcommand*{\@gls@provide@newglossary{%
    \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
    \let\@gls@provide@newglossary\relax
}
\newcommand*{\defglsentryfmt}[2][\glsdefaultttype]{%
    \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{@gls@forbidtexext}[1]{%
    \ifboolexpr{test {\ifdefstring{#1}{tex}}
        or test {\ifdefstring{#1}{TEX}}}
    {%
        \def#1{nottex}%
        \PackageError{glossaries}%
        {Forbidden ‘.tex’ extension replaced with ‘.nottex’}%
        {I’m sorry, I can’t allow you to do something so reckless.\MessageBreak
        Don’t use ‘.tex’ as an extension for a temporary file.}%
    }%
    {%
    }%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{}
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
    \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
    \doifglossarynoexistsordo{#2}%
    {%
        \ifundef\glsdefaultttype
        {%
            \gdef\glsdefaultttype{#2}%
        }{}%
        \toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
        \expandafter\gdef\csname glolist@#2\endcsname{,}%
        \expandafter\edef\csname @gloptype@#2@log\endcsname{#1}%
        \expandafter\edef\csname @gloptype@#2@in\endcsname{#3}%
    }

```

```

\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}{%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxsindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{, #1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%

```

```

{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}%
{}%
\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*{\@gls@nohyperlist}{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*{\@ignored@glossaries}{}
\newcommand*{\ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
    {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*{\@gls@default@value}{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}

```

```

\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
\def\@glo@counter{#1}%
}%
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}%
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}%
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}%
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {‘\gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}%
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
  [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}%
\newcommand*{\@gls@savenonumberlist}[1]{%
\newcommand*{\@gls@initnonumberlist}{%
\newcommand*{\@gls@storenonumberlist}[1]{%

```

```

\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
      {%
        \cslet{glo@glstdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
      }%
      {}%
    }%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@name}{%
  \PackageError{glossaries}{name key required in
    \string\newglossaryentry\space for entry '\@glo@label'}{You
    haven't specified the entry name}}

```

```

\newcommand*\@glsnodedesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@glsdefaultplural{}
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'.
      Maybe makeglossaries + rerun required}%
  \else
    \PackageError{glossaries}%
    {Package option 'savenumberlist=true' required}%
    {%
      You must use the 'savenumberlist' package option
      to reference location lists.%
    }%
  \fi
}
\newcommand*\@glsdefaultsort{}\@glo@name
\newcount\gls@level
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}\@gls@value}%
    }%
    {%
  }%

```



```

        \@gls@noexpand@field{#2}{#3}{#4}%
    }%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*{\glsexpandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \newcommand*{\gls@defdocnewglossaryentry}{%
    \let\newglossaryentry\new@glossaryentry
  }
  \newrobustcmd{\provideglossaryentry}[2]{%
    \ifglsentryexists{#1}%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \@onlypreamble{\provideglossaryentry}
  \newcommand*{\gls@glsdefs@labellist}{}
  \newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
  \newcommand*{\gls@docdef@labellist}{}
  \newcommand*{\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
  \newrobustcmd{\new@glossaryentry}[2]{%
    \ifundef\@gls@deffile
    {%
      \global\newwrite\@gls@deffile
      \immediate\openout\@gls@deffile=\jobname.glsdefs
    }%
    {%
      \ifglsentryexists{#1}%
      {%
        \xifinlist{#1}{\gls@docdef@labellist}%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in the
            document environment (ignoring new definition)}%
        }%
        {%
          \xifinlist{#1}{\gls@glsdefs@labellist}%
          {%
            \gls@defglossaryentry{#1}{#2}%
            \gls@docdef@addlabel{#1}%
            \@gls@writedef{#1}%
          }%
          {%
            \GlossariesWarning{Entry ‘#1’ has already been defined in
              the preamble (in which case remove the duplicate definition)
              or you have just updated to glossaries.sty
              v4.47 (in which case delete ‘\jobname.glsdefs’)}%
          }%
        }%
      }%
    }%
    \gls@defglossaryentry{#1}{#2}%
  }

```

```

\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{\gls@restoreat}{\gls@restoreat}
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}
\newcommand*\gls@defdocnewglossaryentry[2]{%
\ifglsentryexists{#1}%
{\GlossariesWarning{entry '#1' has already been defined in the preamble
(if you have moved the definition there from the document
environment, delete \jobname.glsdefs)}}%
{%
\gls@docdef@addlabel{#1}%
\gls@defdocnewglossaryentry{#1}{#2}%
}%
}
\newcommand*\gls@writedef[1]{%
\immediate\write\@gls@deffile
{%
\string\gls@defdocnewglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
\expandafter\@gobble\string\{\glspercentchar%
}%
\@for\@gls@map:=\@gls@keymap\do
{%
\letcs\gls@value{gls@detoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
\ifdef\gls@value
{%
\@onelevel@sanitize\gls@value
\immediate\write\@gls@deffile
{%
\expandafter\@firstoftwo\@gls@map
=\expandafter\@gobble\string\{\gls@value\expandafter\@gobble\string\},%
\glspercentchar
}%
}%
}%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
\glspercentchar^^J%
}

```

```

\expandafter\@gobble\string\}\glpercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
}%

```

```

\expandafter\newcommand\expandafter*\expandafter
{\csname @gls@user@#1\endcsname}[2] [] {%
  \new@ifnextchar [%
    {\csuse{@gls@user@#1@}{##1}{##2}}}%
    {\csuse{@gls@user@#1@}{##1}{##2} []}}%
\csdef{@gls@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @Gls@user@#1\endcsname}[2] [] {%
  \new@ifnextchar [%
    {\csuse{@Gls@user@#1@}{##1}{##2}}}%
    {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
\csdef{@Gls@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @GLS@user@#1\endcsname}[2] [] {%
  \new@ifnextchar [%
    {\csuse{@GLS@user@#1@}{##1}{##2}}}%
    {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
\csdef{@GLS@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%

```

```

    {%
      \PackageError{glossaries}{Key ‘#1’ already exists}{}%
    }%
  }
\newcommand{\glsfieldxddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
  }%
}

```

```

}%
{%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\glsfieldfetch}[3]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \letcs#3{glo@\@glo@label @#2}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
    \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
    \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
    \fi
}
\newcommand{\ifglsfieldeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\ifglsfielddefeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \expandafter\ifdefstrequal
            \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}

```



```

}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand*{\glswritedefhook}{%
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\@glo@desc}%
\gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bggroup
\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\@glsnname
\let\@glo@desc\@glsnname
\let\@glo@descplural\@gls@default@value
\let\@glo@type\@gls@default@value

```

```

\let\@glo@symbol\@gls@default@value
\let\@glo@symbolplural\@gls@default@value
\let\@glo@text\@gls@default@value
\let\@glo@plural\@gls@default@value
\let\@glo@first\@gls@default@value
\let\@glo@firstplural\@gls@default@value
\let\@glo@sort\@gls@default@value
\let\@glo@counter\@gls@default@value
\def\@glo@see{}%
\def\@glo@parent{}%
\def\@glo@prefix{}%
\@gls@initnonnumberlist
\def\@glo@useri{}%
\def\@glo@userii{}%
\def\@glo@useriii{}%
\def\@glo@useriv{}%
\def\@glo@userv{}%
\def\@glo@uservi{}%
\def\@glo@short{}%
\def\@glo@shortpl{}%
\def\@glo@long{}%
\def\@glo@longpl{}%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glolist@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodelsc
  \let\@glo@desc\@empty
  \fi
}%
{%
}%
}%

```

```

\protected@edef\@glo@list@{\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@glsnoname
        \expandafter\let\expandafter\@glo@name
          \csname glo@\@glo@parent @name\endcsname
        \ifx\@glo@plural\@gls@default@value
          \expandafter\let\expandafter\@glo@plural
            \csname glo@\@glo@parent @plural\endcsname
        \fi
      \fi
    }%
    {%
      \PackageError{glossaries}%
      {%
        Invalid parent ‘\@glo@parent’
        for entry ‘\@glo@label’ - parent doesn’t exist%
      }%
      {%
        Parent entries must be defined before their children%
      }%
      \def\@glo@parent{}%
      \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
    }%
  }%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{\sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{\text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%

```

```

        {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @text\endcsname}%
    {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @plural\endcsname}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
{%
    \def\@glo@defaultcounter{\glscounter}%
}%
{%
    \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
    \@glsnoname
    \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
    \csxdef{glo@\@glo@label @numberlist}{%
        \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
    \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
    \@glsnodesc
    \let\@glo@desc\@gls@default@value

```

```

\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
  \noexpand\global
    \noexpand\let\expandafter\noexpand
      \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
  \noexpand\global
    \noexpand\let\expandafter\noexpand
      \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
  \csdef{glo@\@glo@label @index}{}%
}
{%
  \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
  \ifdefvoid\@glo@see{%
    {%
      \protected@edef\@do@glssee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
      \@do@glssee
    }%
    \@glo@autoseehook
  }%
\newcommand*{\@glo@autoseehook}{%
\newcommand*{\@newglossaryentryprehook}{%
\newcommand*{\@newglossaryentryposthook}{%
\newcommand*{\@newglossaryentry@defcounters}{%
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%

```

```

\protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
\def\glo@list{,}%
\forlssentries[\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}%
      }%
    \fi
  \fi

```

```

\else
\expandafter\protected\def\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

```



```

        {\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{s}format}[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*{\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{spl}format}[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
      {%
        \ifglssused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \ifbool{glscompatible-3.07}%
        {%
          \protected@edef\@glo@etext{%
            #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
          \xmakefirstuc\@glo@etext
        }%
        {%
          #2{\Glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
  }%
}

```

```

{%
\ifbool{glscompatible-3.07}%
{%
\protected@edef\@glo@etext{%
#1{\glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
\xmakefirstuc\@glo@etext
}%
{%
#1{\Glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}%
}%
}%
}%
{%
\ifglsused\glslabel
{%
\mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
{%
\mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
{\glsentrydescplural{\glslabel}}%
{\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
}%
}%
}%
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
{%
#1{\glsentryfirst{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
}%
}%
{%
\ifglsused\glslabel
{%
\ifbool{glscompatible-3.07}%
{%

```

```

\protected@edef\@glo@etext{%
  #2{\glsentrytext{\glslabel}}%
  {\glsentrydesc{\glslabel}}%
  {\glsentrysymbol{\glslabel}}{\glsinsert}}%
\makefirstuc\@glo@etext
}%
{%
  #2{\Glsentrytext{\glslabel}}%
  {\glsentrydesc{\glslabel}}%
  {\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
}%
{%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    \makefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirst{\glslabel}}%
    {\glsentrydesc{\glslabel}}%
    {\glsentrysymbol{\glslabel}}{\glsinsert}%
  }%
}%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}}%
  }%
}%
}%
}%
{%
  \ifglsused{\glslabel}%
  {%
    #2{\glscustomtext}%
    {\glsentrydesc{\glslabel}}%
    {\glsentrysymbol{\glslabel}}{\glsinsert}%
  }%
}

```

```

    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{%}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glusifplural
    {%
      \glscapscase
      {%
        \ifglssused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \ifglssused\glslabel
      {%
        \glsentrytext{\glslabel}\glsinsert
      }%
    }%
  }%
}

```

```

        {%
        \glsentryfirst{\glslabel}\glsinsert
        }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
    \Glsentryfirst{\glslabel}\glsinsert
    }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
    {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
    }%
    }%
    }%
    {%
    \glscustomtext\glsinsert
    }%
}
\newcommand*{\glsnacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\mfirstucMakeUppercase
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
}%
}%
}%
\glscapscase
{%

```



```

        \acronymfont{\glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \mfirstucMakeUppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
}%
}%
}%
{%
    \glscustomtext
}%
}
\newcommand*{\genacrfullformat}[2]{%
    \glsentrylong{#1}#2\space
    (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%

```

```

\protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
\xmakefirstuc\gls@text
}
\newcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\glsdisplayfirst}[4]{#1#4}
\newcommand*{\glsdisplay}[4]{#1#4}
\newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}}%
    {%
      \noexpand\@gls@default@entryfmt
      {\noexpand\csuse{gls@#1@displayfirst}}}%
    {\noexpand\csuse{gls@#1@display}}}%
  }%
  {%
    \noexpand\@gls@default@entryfmt
    {\noexpand\glsdisplayfirst}%
    {\noexpand\csuse{gls@#1@display}}}%
  }%
}%
\@gls@doentrydef
}
\newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}}%
    {%
      \noexpand\@gls@default@entryfmt
      {\noexpand\csuse{gls@#1@displayfirst}}}%
    {\noexpand\csuse{gls@#1@display}}}%
  }%
  {%
    \noexpand\@gls@default@entryfmt
    {\noexpand\csuse{gls@#1@displayfirst}}}%
    {\noexpand\glsdisplay}%
  }%
}

```

```

    }%
  }%
}%
\@gls@doentrydef
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
{%
  \def\@gls@counter{#1}%
}%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
  \glslinkvar{#1}{#2}{#1}%
  \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
    you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3][]{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }
}

```

```

    }{%
      \glstextformat{#3}%
    }%
    \glspostlinkhook
  }
  \newcommand*{\glspostlinkhook}{}
  \newcommand*{@gls@link@checkfirsthyper}{%
    \ifglsused{\glslabel}%
    {%
    }%
    {%
      \gls@checkisacronymlist\glstype
      \ifglshyperfirst
        \if@glsisacronymlist
          \ifglsacrfootnote
            \KV@glslink@hyperfalse
          \fi
        \fi
      \else
        \KV@glslink@hyperfalse
      \fi
    }%
    \glslinkcheckfirsthyperhook
  }
  \newcommand*{\glslinkcheckfirsthyperhook}{}
  \newcommand*{\glslinkpostsetkeys}{}
  \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
  \newcommand*{\do@glsdisablehyperinlist}{%
    \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
    {\KV@glslink@hyperfalse}}%
  }
  \newcommand*{@gls@setdefault@glslink@opts}{}
  \def\@gls@link[#1]#2#3{%
    \leavevmode
    \protected@edef\glslabel{\glsdetoklabel{#2}}%
    \def\@gls@link@opts{#1}%
    \let\@gls@link@label\glslabel
    \def\@glsnumberformat{\glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
    \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
    \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
    \@gls@setdefault@glslink@opts
    \do@glsdisablehyperinlist
    \do@gls@link@checkfirsthyper
    \setkeys{glslink}{#1}%
    \glslinkpostsetkeys
    \@gls@saveentrycounter
    \@gls@setsort{\glslabel}%
    \@do@wrglossary{#2}%
    \ifKV@glslink@hyper

```

```

        \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \else
        \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
    \fi
    \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{@gls@saveentrycounter}{%
    \def{@gls@Hcounter}{%
        \ifthenelse{\equal{@gls@counter}{equation}}{%
            {
                \ifcsundef{xatlevel@}%
                {%
                    \edef\theglsentrycounter{\expandafter\noexpand
                        \csname the{@gls@counter}\endcsname}%
                }%
                {%
                    \ifx\xatlevel@\@empty
                        \edef\theglsentrycounter{\expandafter\noexpand
                            \csname the{@gls@counter}\endcsname}%
                    \else
                        \savecounters@
                        \advance\c@equation by 1\relax
                        \edef\theglsentrycounter{\csname the{@gls@counter}\endcsname}%
                        \ifcsundef{theH{@gls@counter}}%
                        {%
                            \def{@gls@Hcounter}{\theglsentrycounter}%
                        }%
                        {%
                            \def{@gls@Hcounter}{\csname theH{@gls@counter}\endcsname}%
                        }%
                        \protected@edef\theHglentrycounter{@gls@Hcounter}%
                        \restorecounters@
                    \fi
                }%
            }%
        }%
        \edef\theglsentrycounter{\expandafter\noexpand
            \csname the{@gls@counter}\endcsname}%
    }%
    \ifx{@gls@Hcounter}\@empty
        \ifcsundef{theH{@gls@counter}}%
        {%
            \def\theHglentrycounter{\theglsentrycounter}%
        }%
        {%
            \protected@edef\theHglentrycounter{\expandafter\noexpand
                \csname theH{@gls@counter}\endcsname}%
        }%
    }%
}

```

```

\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1(\relax
\def\@glo@prefix{)%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
}

```

```

\gls@dostsubst
}%
\let#1=\gls@xdyststring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifglxsindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescactual#1\@nil"??"\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkesclevel#1\@nil\!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\edef\@do@gls@checkesc{%
\noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkquote{\relax}%

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkescquote{\@gls@checkescquote#2""\null}%
\else
\def\@@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar

```



```

        \@gls@quotechar\string\" \@gls@actualchar}%
        \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
    \fi
\else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\string\" \@gls@actualchar}%
    \ifx\null#3\null
        \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
    \else
        \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
    \fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkescbar{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string\" \@gls@encapchar
            \@gls@quotechar\string\" \@gls@encapchar}%
            \def\@gls@checkescbar{\@gls@checkescbar#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string\" \@gls@encapchar}%
        \ifx\null#3\null
            \def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
        \else
            \def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
        \fi
    \fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkesclevel{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string\" \@gls@levelchar
            \@gls@quotechar\string\" \@gls@levelchar}%
            \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
        \fi
    \fi
\@gls@checkesclevel
}

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\"@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%

```

```

\ifx\null#3\null
  \def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
  \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesc{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
      \def\@gls@checkesc{\@gls@checkesc#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@backslashchar}%
    \ifx\null#3\null

```

```

        |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
    |else
        |def|@gls@checkesc{|@gls@checkesc#2\#3|null}%
    |fi
|fi
|@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@xdycheckquote{\relax}%
        |else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \string"\string"}%
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
        \fi
    |else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \string"}%
        \ifx\null#3\null
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
        |else
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
        \fi
    \fi
    \@gls@xdycheckquote
}
\edef\def\@gls@xdycheckbackslash{%
    \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
        ##2\@backslashchar##3\noexpand\null{%
        \noexpand\@gls@tmpb=\noexpand\expandafter
            {\noexpand\@gls@checkedmkidx}%
        \noexpand\toks@={##1}%
        \noexpand\ifx\noexpand\null##2\noexpand\null
            \noexpand\ifx\noexpand\null##3\noexpand\null
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
                \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
            \noexpand\else
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
                    \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
            \noexpand\def\noexpand\@gls@xdycheckbackslash{%
                \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
            \noexpand\fi
        \noexpand\else

```

```

\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    \@backslashchar\noexpand\null}%
\noexpand\else
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    ##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength@gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
  \glsshowtarget{#1}%
  \settoheight{\gls@tmplen}{#2}%
  \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*{\glsdohyperlink}[2]{%
  \glsshowtarget{#1}%
  \hyperlink{#1}{#2}%
}
\newcommand*{\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue

```

```

\let\@glslink\glsdohyperlink
\let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}

```

```

\newcommand*{\@GLS}[2][]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glS@link@checkfirsthyper\@glS@link@checkfirsthyper
    \let\glSifplural\@secondoftwo
    \let\glScapscase\@thirdofthree
    \let\glScustomtext\@empty
    \def\glSinsert{#3}%
    \def\@glo@text{\csname glS@\glstype @entryfmt\endcsname}%
    \@glS@link[#1]{#2}{\@glo@text}%
    \ifKV@glSlink@local
      \glSlocalunset{#2}%
    \else
      \glSunset{#2}%
    \fi
  }%
  \glSpostlinkhook
}
\newrobustcmd*{\glSpl}{\@glS@hyp@opt\@glSpl}
\newcommand*{\@glSpl}[2][]{%
  \new@ifnextchar[{\@glSpl@{#1}{#2}}{\@glSpl@{#1}{#2}[]}%
}
\def\@glSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glS@link@checkfirsthyper\@glS@link@checkfirsthyper
    \let\glSifplural\@firstoftwo
    \let\glScapscase\@firstofthree
    \let\glScustomtext\@empty
    \def\glSinsert{#3}%
    \def\@glo@text{\csname glS@\glstype @entryfmt\endcsname}%
    \@glS@link[#1]{#2}{\@glo@text}%
    \ifKV@glSlink@local
      \glSlocalunset{#2}%
    \else
      \glSunset{#2}%
    \fi
  }%
  \glSpostlinkhook
}
\newrobustcmd*{\Glspl}{\@glS@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\gl@sifplural\@firstoftwo
\let\gl@scapscase\@secondofthree
\let\gl@customtext\@empty
\def\gl@insert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link[#1]{#2}{\@glo@text}%
\ifKV@gl@link@local
  \gl@localunset{#2}%
\else
  \gl@unset{#2}%
\fi
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else
      \gl@unset{#2}%
    \fi
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@doifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else

```



```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link[#1]{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
\newcommand*{\@glstext}[2][]{%
    \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
\def\@glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glstentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\newcommand*{\@Glstext}[2][]{%
    \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
\def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][]{%
    \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
\def\@glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glstentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][]{%
    \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}}
\def\@Glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][]{%
    \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]}}
\def\@GLSfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentryfirst{#2}#3}}%
}
}

```

```

\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2][]{\%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2}[]}}
\def\@glsplural@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2][]{\%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}}
\def\@Glsplural@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2][]{\%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2}[]}}
\def\@GLSplural@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][]{\%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}}
\def\@glsfirstplural@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][]{\%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}}
\def\@Glsfirstplural@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2][]{\%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2}[]}}
\def\@GLSfirstplural@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2][]{\%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2}[]}}
\def\@glsname@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2][]{\%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2}[]}}
\def\@Glsname@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2][]{\%

```

```

\new@ifnextchar [{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] [] {%
  \new@ifnextchar [{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2] [] {%
  \new@ifnextchar [{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar [{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar [{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
  \new@ifnextchar [{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}}
\def\@Glsymbol@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2][]{%
    \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2}[]}}
\def\@GLSsymbol@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2][]{%
    \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2}[]}}
\def\@glssymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2][]{%
    \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2}[]}}
\def\@Glsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2][]{%
    \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2}[]}}
\def\@GLSsymbolplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2][]{%
    \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2}[]}}
\def\@glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][]{%
    \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}}
\def\@Glsuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][]{%
    \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
\def\@GLSuseri@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][]{%
    \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
\def\@glsuserii@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}

```



```

\new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2] [] {%
  \new@ifnextchar [{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}}
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] [] {%
  \new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\newcommand*{\@Glsuservi}[2] [] {%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}}
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\@ns@acrshort}[2] [] {%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}}
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
}

```

```

\glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\Glsentryshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo

```

```

\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\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@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@firstofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \glsl@entrylong{#2}#3%
    }%
    \@glsl@link{#1}{#2}{\csname glsl@glstype @entryfmt\endcsname}%
  }%
  \glsl@postlinkhook
}
\newrobustcmd*{\Acrlong}{\@glsl@hyp@opt\@ns@Acrlong}
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@secondofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \glsl@entrylong{#2}#3%
    }%
    \@glsl@link{#1}{#2}{\csname glsl@glstype @entryfmt\endcsname}%
  }%
  \glsl@postlinkhook
}
\newrobustcmd*{\ACrlong}{\@glsl@hyp@opt\@ns@ACrlong}
\newcommand*{\ns@ACrlong}[2][{}]{%
  \new@ifnextchar[{\@ACrlong{#1}{#2}}{\@ACrlong{#1}{#2}[]}%
}
\def\@ACrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@thirdofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \mfirstucMakeUppercase{\glsl@entrylong{#2}#3}%
    }%
  }%
}

```

```

    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\@ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glstrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\AcrLongpl}{\@gls@hyp@opt\@ns@AcrLongpl}
\newcommand*{\ns@AcrLongpl}[2][]{%
  \new@ifnextchar[{\@AcrLongpl{#1}{#2}}{\@AcrLongpl{#1}{#2}[]}%
}
\def\@AcrLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRLongpl}{\@gls@hyp@opt\@ns@ACRLongpl}
\newcommand*{\ns@ACRLongpl}[2][]{%
  \new@ifnextchar[{\@ACRLongpl{#1}{#2}}{\@ACRLongpl{#1}{#2}[]}%
}
\def\@ACRLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@\#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@\#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@\#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ??%
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  }%
  \else

```

```

\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newrobustcmd*{\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*{\glsentrydescplural}[1]{%
\@gls@entry@field{#1}{descplural}%
}
\newrobustcmd*{\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newrobustcmd*{\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*{\glsentryplural}[1]{%
\@gls@entry@field{#1}{plural}%
}
\newrobustcmd*{\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*{\glsentrysymbol}[1]{%
\@gls@entry@field{#1}{symbol}%
}
\newrobustcmd*{\Glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}

```

```

}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*{\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*{\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*{\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newcommand*{\glscapitalisewords}[1]{%
  \@capitalisewords{#1}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsentrytitlecase}[2]{%
    \texorpdfstring
    {\@glsentrytitlecase{#1}{#2}}%
    {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*{\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}

```

```

\newrobustcmd*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*{\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*{\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*{\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*{\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*{\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*{\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*{\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*{\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*{\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*{\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*{\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*{\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*{\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
\newcommand*{\glsentryfull}[1]{%

```

```

\acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
\acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
\acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
\acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\@gls@entry@field{#1}{numberlist}%
}%
}
\@ifpackageloaded{hyperref}{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\GlossariesWarning
{%
\string\glsdisplaynumberlist\space
doesn't work with hyperref.^^JUsing
\string\glsentrynumberlist\space instead%
}%
\glsentrynumberlist{#1}%
}%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\bgroup
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\@org@glsnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
\csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
\protected@edef\@gls@thislist{%
\@gls@thislist
\noexpand\@gls@numlist@sep
##1%
}%
\let\@gls@numlist@sep\@gls@numlist@nextsep

```

```

\def\@gls@numlist@nextsep{\glsnumlistsep}%
\@gls@donext@def
\def\@gls@donext@def{%
\def\@gls@numlist@lastsep{\glsnumlistlastsep}%
}%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
\def\@glo@label{#2}%
\@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][ ]{%
\@gls@adjustmode
\glsdoifexists{#2}%
{%
\def\@glsnumberformat{glsnumberformat}%
\protected@edef\@gls@counter{\csname glo@%glsdetoklabel{#2}@counter\endcsname}%
\setkeys{glossadd}{#1}%
\@gls@saveentrycounter
\@gls@setsort{#2}%
\@do@wrglossary{#2}%
}%
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][ ]{%
\protected@edef\@glo@type{\@glo@types}%
\setkeys{glossadd}{#1}%
\forallglsentries[\@glo@type]{\@glo@entry}{%
\glsadd[#1]{\@glo@entry}%
}%
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
\forallglsentries[#1]{\@glo@entry}%
{%
\ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
}%
}
\newcommand*{\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{}}

```



```

\edef\glsclosebrace{\expandafter\@gobble\string\}}
\edef\glslbackslash{\expandafter\@gobble\string\}
\edef\glquote#1{\string"#1\string"}
\edef\glpercentchar{\expandafter\@gobble\string\%}
\edef\glstildechar{\string~}
\ifglxindy
  \newcommand*{\@glsfirstletter}{A}
\fi
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglxindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder}
\ifglxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnxindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%

```

```

\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglxindy
\def\writeist{%
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \@gls@addpredefinedattributes
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{{(define-location-class
      \string"\@gls@classI\string"^^J\space\space\space
      (
        :sep "{{"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
      )
    }}%
  }%
  \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{{(define-location-class
      \string"\@gls@classII-\@gls@classI\string"
      ^^J\space\space\space
      (
        :sep "{"
        \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
        :sep "{{"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
    }}%
  }%

```

```

        ^^J\space\space\space
        :min-range-length \@glxminrange^^J%
    )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glssseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{}\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}%
\write\glswrite{(markup-letter-group-list

```

```

:sep \string"\string\glsgroupskip\glstildechar n\string"))}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string"))}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"))}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string"))}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string"))}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"))}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string"))}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glpercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glpercentchar\space for document
  'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@glactualchar'}
\write\glswrite{encap '@glencapchar'}
\write\glswrite{level '@gllevelchar'}
\write\glswrite{quote '@glquotechar'}
\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\glresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string\}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\}\string}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glresetentrylist \string}
\write\glswrite{symhead_positive \string\glssymbols\string}
\write\glswrite{numhead_positive \string\glnumbers\string}
\write\glswrite{page_compositor \string\glscpositor\string}
\@glscbsdq\glscsuffixF

```

```

\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{ }
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%

```

```

\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
\noexpand\\|noexpand\\|noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand|string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand|string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand|string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null}%
\noexpand\fi

```



```

        \noexpand\fi
        \noexpand\@@gls@checkescquote
    }%
}%
\@gls@docheckquotedef
}
\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as
    soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*{\@gls@extramakeindexopts}[1]{%
\newcommand{\noist}{%
    \@gls@addpredefinedattributes
    \let\writeist\relax
}
\newcommand*{\@makeglossary}[1]{%
    \ifglossaryexists{#1}%
    {%
        \ifglssavewrites
            \expandafter\newtoks\csname glo@#1@filetok\endcsname
        \else
            \expandafter\newwrite\csname glo@#1@file\endcsname
            \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
        \fi
        \@gls@renewglossary
        \writeist
    }%
    {%
        \PackageError{glossaries}%
        {Glossary type ‘#1’ not defined}%
        {New glossaries must be defined before using \string\makeglossaries}%
    }%
}
\newcommand*{\@glsopenfile}[2]{%
    \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
        \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
}
\ifglsxindy
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-M ord/letorder }}%
        {\let\@gls@order\@empty}%
    }

```

```

\ifcsundef{@xdy@#1@language}%
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilename\space
-C \@gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilebase\space
-C \@gls@codepage\space
-t \jobname.\csuse{\@glotype@#1@log}
-o \jobname.\csuse{\@glotype@#1@in}
\jobname.\csuse{\@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{\@glotype@#1@out}}%
{%
\ifdefstring{\@gls@order}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{\@glotype@#1@log}
-o \jobname.\csuse{\@glotype@#1@in}
\jobname.\csuse{\@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\@gls@automake@nr=2\relax

```

```

\@for\@gls@type:=\@glo@types\do{%
  \ifempty{\@gls@type}{}%
  {\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
  \@gls@@automake@immediate
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
  \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
  \ifundef\@gls@extramakeindexopts
  {}%
  {%
    \protected@write\@auxout{}{\string\providecommand
      \string\@gls@extramakeindexopts[1]{} }
    \protected@write\@auxout{}{\string\@gls@extramakeindexopts
      {\@gls@extramakeindexopts}}%
  }%
  \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
  \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
  \@for\@glo@type:=\@glo@types\do{%
    \ifthenelse{\equal{\@glo@type}{}}{ }{%
      \@makeglossary{\@glo@type}}%
    }%
  \renewcommand*\newglossary[4][]{%
  \PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
  \let\@makeglossary\@gobble
  \let\makeglossaries\relax
  \@disable@onlypremakeg
  \let\gls@checkseeallowed\relax
  \let\warn@nomakeglossaries\relax
  \let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
  \iflssavenumberlist
    \edef\@gls@dodeflistparser{\noexpand\DeclareListParser
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodeflistparser
  \fi
  \let\makenoidxglossaries\@no@makeglossaries
  \renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
  }%
  \ifglsautomake

```

```

\renewcommand*{\@gls@doautomake}{%
  \@for\@gls@type:=\@glo@types\do{%
    \ifdefempty{\@gls@type}{}%
    {\@gls@automake{\@gls@type}}%
  }%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}}%
      {%
        \string\glsseeformat##2}%
    }%
  }%
}

```

```

    }%
  }%
}%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{\glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{\glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
  }%

```

```

    }%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

```

\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
    {Either use one or other (or none) of those commands but not both
    together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}}%
      }%
  }%
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \foralllglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%

```



```

        \expandafter\the
        \csname glo@%glo@type @filetok\endcsname}%
    \immediate\closeout\glswrite
\fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
            }%
            {%
                \ifignoredglossary{#1}{}%
                {%
                    \GlossariesWarning{No file defined for glossary ‘#1’}%
                }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*{\gls@writeentry}[2]{%
    \ifglsindexonlyfirst
        \ifglsused{#1}{#2}%
    \else
        #2%
    \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
    \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

```

\newcommand*{\gls@disablepagerefexpansion}{%
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
}
\newcommand*{\gls@alphapage}{\@alph\c@page}
\newcommand*{\gls@Alphapage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\@number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\@romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\gls@saddprotectedpagefmt}[1]{%
  \protected@eappto\gls@protected@pagefmts{\, \expandonce{\csname gls#1page\endcsname}}%
  \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
  \protected@eappto\@wrglossarynumberhook{%
    \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
    \expandonce{\csname#1\endcsname}%
    \noexpand\def\expandonce{\csname#1\endcsname}{%
      \noexpand\@wrglossary@pageformat
      \expandonce{\csname gls#1page\endcsname}%
      \expandonce{\csname org@gls#1\endcsname}%
    }%
  }%
}
\newcommand*{\@wrglossarynumberhook}{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@@do@wrglossary}[1]{%
  \ifglsclocations
    \@@do@esc@wrglossary{#1}%
  \else
    \@@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*{\@@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theHglentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@@do@@wrglossary
}

```

```

\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
\beginingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\protected@edef\@gls@label{\gls@detoklabel{#1}}%
\@@do@@wrglossary
}
\newcommand*{\@@do@@wrglossary}{%
\ifglxindy
\expandafter\@glo@check@mkidxrangechar\@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)
:locoref \string"\@glo@counterprefix}{\@gls@locoref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locoref}}%
\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
\edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been^^J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref

```

```

\ifglxsindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
  \fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat}[1]{\glstrytext{#1}}
\newcommand*\gls@save@numberlist}[1]{%
  \ifglssavenumberlist

```

```

\toks@{#1}%
\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\auxout{}\do@writeaux@info{%
\fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
\newcommand*{\@gls@preglossaryhook}{%
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

```

```

\def\@glossarystyle{%
  \ifx\@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '\@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def\gls@dotoc@title{\gls@settoc@title{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
  {%
    \ifx\glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary\@glo@type
    \let\org@glossaryentrynumbers\glossaryentrynumbers
    \let\glsnonextpages\@glsnonextpages
    \let\glsnextpages\@glsnextpages
    \let\nopostdesc\@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \protected@xdef\gls@currententrylabel{\gls@detoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \protected@xdef\gls@currententrylabel{\gls@detoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\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}%
\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@=\uccode' #1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@=' #1\relax
}%
{%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{\glo@\glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{\glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
{%
\subglossentry{\gls@level}{#1}%
{%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%

```

```

    }%
  }%
}%
{%
  \letcs{\@gls@sort}{glo@glstetoklabel{#1}@sort}%
  \expandafter\glo@grabfirst\@gls@sort{}\{}@nil
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \global\let\@glo@thislettergrp\@glo@thislettergrp
      \glsgrpskip
    }%
    \glsgrpskipheading{\@glo@thislettergrp}%
  }%
  \global\let\@gls@currentlettergroup\@glo@thislettergrp
  \ifdefvoid{\@gls@loclist}
  {%
    \glossentry{#1}{}%
  }%
  {%
    \glossentry{#1}%
    {%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}%
}
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%

```

```

\@gls@noidxloclist@sep
\@gls@noidxloclist@prev
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
\setentrycounter[#1]{#2}%
\csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
\glsdoifexistsorwarn{#2}%
{%
\ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
\ifinlistcs{#2}{\@glsref@#1}%
{}%
{\listcsgadd{\@glsref@#1}{#2}}%
\ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
{\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
{}%
\listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
}%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{\@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossentrycompatibility
\csname @glsstyle@#1\endcsname}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\@gls@numberedsection@nr\relax
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{}%
\or

```

```

\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
}
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*{\@glsnextpages}{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```



```

\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglsentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglsentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

```

\newcommand*{\glsentryitem}[1]{%
  \ifglsentrycounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*{\glsesubentryitem}[1]{%
  \ifglsesubentrycounter
    \glsstepsubentry{#1}\glsesubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\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@#1\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}

```

```

    }%
  }
  \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \providecommand*{\compatiblesubglossentry}[3]{%
    \toks@{#3}%
    \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}%
      {\noexpand\glsnamefont
        {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
      {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
      {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
      {\the\toks@}%
    }%
    \@do@subglossentry
  }
  \newcommand*{\setglossentrycompatibility}{%
    \let\glossentry\compatibleglossentry
    \let\subglossentry\compatiblesubglossentry
  }
  \setglossentrycompatibility
  \newcommand{\glossaryentryfield}[5]{%
    \GlossariesWarning
    {Deprecated use of \string\glossaryentryfield.^J
      I recommend you change to \string\glossentry.^J
      If you've just upgraded, try removing your gls auxiliary
      files^^J and recompile}%
    \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
  \newcommand*{\glossarysubentryfield}[6]{%
    \GlossariesWarning
    {Deprecated use of \string\glossarysubentryfield.^J
      I recommend you change to \string\subglossentry.^J
      If you've just upgraded, try removing your gls auxiliary
      files^^J and recompile}%

```

```

\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
  \dtl@ifsingle{#1}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
  }%
  {%
    \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
      or test{\ifstrequal{#1}{glsnumbers}}}%
    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
    }%
    {%
      \def#2{#1}%
    }%
  }%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
  }%
}
\newcommand*{\glsgrouplabel}[1]{%
\ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
\ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsetentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
     I recommend you switch to \string\setglossarystyle\space unless
     you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
}

```

```

\def\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\@gls@x{\the\toks@}%
  \ifx\@gls@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{\glshypernumber{#1}}}
\newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}
\newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
\newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}
\newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}
\newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
\newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
\newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
\newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{\short}
\newcommand*{\glsshortpluralkey}{\shortplural}
\newcommand*{\glslongkey}{\long}
\newcommand*{\glslongpluralkey}{\longplural}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}{%
    {\@acrfull{#1}{#2}[]}%
  }
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][{}]{%

```

```

\new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2} []}%
}
\def\@Acrfull#1#2[#3]{%
    \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
    \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*{\ns@ACRfull}[2] []{%
    \new@ifnextchar[{\@ACRfull{#1}{#2}}%
        {\@ACRfull{#1}{#2} []}%
}
\def\@ACRfull#1#2[#3]{%
    \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*{\ns@acrfullpl}[2] []{%
    \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
        {\@acrfullpl{#1}{#2} []}%
}
\def\@acrfullpl#1#2[#3]{%
    \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*{\ns@ACRfullpl}[2] []{%
    \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
        {\@ACRfullpl{#1}{#2} []}%
}
\def\@ACRfullpl#1#2[#3]{%
    \ACRfullplfmt{#1}{#2}{#3}%
}
}

```



```

\newcommand*{\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \gslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
}

```

```

\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{\acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDisplayStyle{#1}}%
    }%
  }%
}
\newcommand*{\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
\newcommand*{\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%

```

```

\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glstrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glstryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glstrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}

```

```

}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})%}
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{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}%
{%
\GlsUseAcrEntryDisplayStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}

```

```

}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%

```

```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink{##1}{##2}{\glsentrylong{##2}##3\space

```

```

        (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
            (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
        (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
            (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
    \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
    \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
    \GlsUseAcrEntryDisplayStyle{dua}%
}%
{%
    \GlsUseAcrStyleDefs{dua}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
    \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%

```



```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
    (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%

```

```

}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
}

```

```

\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\aclong
  \let\aclp\aclongpl
  \let\Acl\Aclong
  \let\Aclp\Aclongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\newcommand*{\@gl@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\glslabeltok @plural\endcsname\@gl@@default@acr@plural
    \csxdef{glo@the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@the\glslabeltok @firstpl\endcsname\@gl@@default@acr@firstplural
    \csxdef{glo@the\glslabeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
    \fi
  \expandafter\ifx\csname glo@the\glslabeltok @descplural\endcsname\@gl@@default@acr@descplural
    \csxdef{glo@the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
    \fi
}
\newcommand*{\@gl@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@gl@@default@acr@firstplural}{%
  \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@gl@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gl@type}%
  }%
}
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@gl@acronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDefaultAcronymDisplayStyle{\@glo@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook

```

```

\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrnolinkfootnote}[3]{%
\footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\glsentryfmt}%
}%
{%
\firstacronymfont{\glsentryfmt}%
\ifglsymbol{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
{%
\glsifplural
{\glsentrysymbolplural{\glslabel}}%
{\glsentrysymbol{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%

```

```

        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        symbol={\the\glslongtok},%
        symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
    \ifglsacrsmallcaps
        \@gls@deprecated@acr@warning{footnote,smallcaps,description}%
        {footnote-sc-desc}{short-sc-footnote-desc}%
    \else
        \ifglsacrsmaller
            \@gls@deprecated@acr@warning{footnote,smaller,description}%
            {footnote-sm-desc}{short-sm-footnote-desc}%
        \else
            \@gls@deprecated@acr@warning{footnote,description}%
            {footnote-desc}{short-footnote-desc}%
        \fi
    \fi
\renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
        \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
}%

```

```

\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
  \defglsentryfmt[##1]{\glsentryfmt}%
}
\newcommand*{\DescriptionDUANewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\ifglsacrsmaller
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
\@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
}%
\glsgenentryfmt
\ifglshassymbol{\glslabel}%
{%
\glsifplural

```



```

        {%
        \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}}%
        }%
        {%
        \def\@glo@symbol{\glsentrysymbol{\glslabel}}}%
        }%
        \space(\protect\firstacronymfont
        {\glscapscase
        {\@glo@symbol}
        {\@glo@symbol}
        {\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
    }%
    }%
    {\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
sort={\the\glsshorttok},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%

```

```

\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{smallcaps,description}%
{long-sc-short-desc}{long-short-sc-desc}%
\else
\ifglsacrsmaller
\@gls@deprecated@acr@warning{smaller,description}%
{long-sm-short-desc}{long-short-sm-desc}%
\else
\@gls@deprecated@acr@warning{description}%
{long-short-desc}{long-short-desc}%
\fi
\fi
\renewcommand{\newacronym}[4][\]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty

```

```

\ifglused{\glslabel}%
{%
  \acronymfont{\glsgenentryfmt}\gls@org@insert
}%
{%
  \firstacronymfont{\glsgenentryfmt}\gls@org@insert
  \ifglshaslong{\glslabel}%
  {%
    \expandafter\protect\expandafter\acrfootnote\expandafter
    {\@gls@link@opts}{\@gls@link@label}%
    {%
      \glsifplural
        {\glsentrylongpl{\glslabel}}%
        {\glsentrylong{\glslabel}}%
    }%
  }%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\the\glslongtok},%
    descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetFootnoteAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
    \else
      \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
    \fi
  \fi
\renewcommand{\newacronym}[4][ ]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetFootnoteAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \FootnoteNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
    can’t both be set}{}%
\fi
}%
\DeclareRobustCommand*\glsdoparenifnotempty}[2]{%

```

```

\protected@edef\gls@tmp{#1}%
\ifdefempty\gls@tmp
{}%
{%
  \ifx\gls@tmp\@gls@default@value
  \else
    \space (#2{#1})%
  \fi
}%
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\gls@customtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\gls@genentryfmt}\gls@org@insert
      }%
      {%
        \gls@genentryfmt
        \ifgls@hasymbol{\glslabel}%
        {%
          \gls@ifplural
          {%
            \def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
          }%
          \space
          (\gls@scapscase
          {\firstacronymfont{\@glo@symbol}}%
          {\firstacronymfont{\@glo@symbol}}%
          {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\gls@customtext\glsinsert}%
  }%
}
\newcommand*{\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\gls@shorttok}},%
      sort={\the\gls@shorttok},%
    }%
  }%
}

```

```

text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
  \else
    \@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
  \fi
\renewcommand{\newacronym}[4][\]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi

```

```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
}%

```

```

\let\@org@gl@assign@firstpl\gl@assign@firstpl
\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@symbolplural\gl@assign@symbolplural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\let\gl@assign@descplural\@org@gl@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \@gl@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@acronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\@gl@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
    Use \string\setacronymstyle{#2} with just glossaries.sty or
    use \string\setabbreviationstyle[acronym]{#3} with
    glossaries-extra.sty}%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle

```



```

\ifglsacrdescription
\ifglsacrfootnote
\SetDescriptionFootnoteAcronymStyle
\else
\ifglsacrdua
\SetDescriptionDUAAcronymStyle
\else
\SetDescriptionAcronymStyle
\fi
\fi
\else
\ifglsacrfootnote
\SetFootnoteAcronymStyle
\else
\ifthenelse{\boolean{glsacrsmallcaps}\OR
\boolean{glsacrsmaller}}{%
}%
\SetSmallAcronymStyle
}%
}%
\ifglsacrdua
\SetDUASyle
\fi
}%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
\defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
name={\the\glsshorttok},%
description={\the\glslongtok},%
first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
firstplural={\acrfullformat
{\noexpand\glsentrylongpl{\the\glslabeltok}}%
{\noexpand\glsentryshortpl{\the\glslabeltok}}},%
text={\the\glsshorttok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
\protected@edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

        user1={\the\glsshorttok},%
        user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
        user3={\the\glslongtok},%
        user4={\the\glslongtok\noexpand\acrpluralsuffix},%
        \CustomAcronymFields,%
        \the\glskeylisttok
    }%
}%
\@do@newglossaryentry
}
\newcommand*{\SetCustomStyle}{%
    \renewcommand{\newacronym}[4][{}]{%
        \ifx\@glsacronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetCustomDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \CustomNewAcronymDef
    }%
    \@for\@gls@type:=\@glsacronymlists\do{%
        \SetCustomDisplayStyle{\@gls@type}%
    }%
}
\RequirePackage{glossary-hypernav}[=v4.49]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
    \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}

```

```

\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showgloftype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}

```

```

}
\newcommand*{\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \cshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*{\showglossaries}{%
  \show\@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}

```

```

{%
\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The 'glossaries-accsupp'
package has been loaded\MessageBreak
after the 'glossaries-extra' package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the 'accsupp' option when you load\MessageBreak
'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
before loading 'glossaries-extra'}%
\fi
}
{}
\def\compatibleglossentry#1#2{%
\toks@{#2}%
\protected@edef\@do@glossentry{%
\noexpand\accsuppglossaryentryfield{#1}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\@do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@subglossentry
}
\RequirePackage{glossaries}[v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\gls@accessibility}[4][]{%
\@gls@showaccsupp{#1}{#2}{#3}%

```

```

\gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
  \def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
  \def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%
  \def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
  \def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
  \def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
  \def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
  \def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
  \def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
  \def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
  \def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
  \def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
  \def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
  \def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
  \def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
  \def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
  \def\@glo@useriiiaccess{#1}%
}
}

```

```

\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaaccess},%
  {user3access}{useriiiaaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaaccess{\relax}%
  \def\@glo@useriiiaaccess{\relax}%
}

```

```

\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
\@gls@oldnewglossaryentryposthook
\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
\@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
\@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
\@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
\@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
\@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
\@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
\@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
\@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
\@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
\@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
\@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
\@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
\@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
\@glo@useriaccess}%
\expandafter

```



```

\protected@xdef\csname glo@\glo@label @useriiaccess\endcsname{%
  \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}
\newcommand*{\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*{\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

```

}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glstr\glscategory{#4}#3accsupp}%
    {\csname glstr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glstr\glscategory{#4}accsupp}%
      {\csname glstr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}

```

```

\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*{\glstextaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%

```

```

{#1}%
{\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@descaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@userivaccess}%

```

```

    {#1}%
    {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
  }
  \newrobustcmd*{\glsuservaccessdisplay}[2]{%
    \ifcsundef{gls@glstoklabel{#2}@uservaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
  }
  \newrobustcmd*{\glsuserviaccessdisplay}[2]{%
    \ifcsundef{gls@glstoklabel{#2}@userviaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
  }
  \newrobustcmd*{\glsaccessdisplay}[3]{%
    \ifcsundef{gls#1accessdisplay}%
    {%
      \PackageError{glossaries-accsupp}{No accessibility support
        for key ‘#1’}{}%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
  }
  \renewcommand*{\@gls@default@entryfmt}[2]{%
    \ifdefempty{glscustomtext}
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \ifglsused{glslabel}
          {%
            #2{\glspluralaccessdisplay
              {\glsentryplural{\glslabel}}{\glslabel}}%
              {\glsdescriptionpluralaccessdisplay
                {\glsentrydescplural{\glslabel}}{\glslabel}}%
              {\glsymbolpluralaccessdisplay
                {\glsentrysymbolplural{\glslabel}}{\glslabel}}
              {\glsinsert}%
            }%
          }%
          #1{\glsfirstpluralaccessdisplay
            {\glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
              {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glsymbolpluralaccessdisplay
              {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
          }%
        }%
      }%
    }%
  }

```

```

{%
  \ifglused\glslabel
  {%
    #2{\glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
{%
  \ifglused\glslabel
  {%
    \MakeUppercase{%
      #2{\glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
        {\glssymbolpluralaccessdisplay
          {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
      }%
    }%
    \MakeUppercase{%
      #1{\glsfirstpluralaccessdisplay
        {\glsentryfirstplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
        {\glssymbolpluralaccessdisplay
          {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
      }%
    }%
  }%
}%
\glscapscase
{%
  \ifglused\glslabel
  {%

```

```

#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}

```

```

}%
{%
  \MakeUppercase{%
    #1{\glsfirstaccessdisplay
      {\glsentryfirst{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glssymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
  }%
}%
}%
}%
{%
  \ifglsused{\glslabel}%
  {%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glssymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glssymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}
\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glsfirstpluralaccessdisplay
            {\glsentryfirstplural{\glslabel}}{\glslabel}%

```



```

\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
\glsfirstpluralaccessdisplay
{\Glsentryfirstplural{\glslabel}}{\glslabel}%
\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\glspluralaccessdisplay
{\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
{\glslabel}%
\mfirstucMakeUppercase{\glsinsert}%
}%
{%
\glsfirstpluralacessdisplay
{\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
{\glslabel}%
\mfirstucMakeUppercase{\glsinsert}%
}%
}%
}%
{%
\glscapscase
{%
\ifglsused\glslabel
{%
\glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
\glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
}%
}%
{%
\ifglsused\glslabel
{%
\glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
\glsinsert

```

```

}%
{%
  \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
  \glsinsert
}%
}%
{%
  \ifglsused\glslabel
  {%
    \glstextaccessdisplay
    {\mfirstucMakeUppercase{\glsentrytext{\glslabel}}}{\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstaccessdisplay
    {\mfirstucMakeUppercase{\glsentryfirst{\glslabel}}}{\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \mfirstucMakeUppercase
          {\acronymfont
            {\glsshortpluralaccessdisplay
              {\glsentryshortpl{\glslabel}}{\glslabel}}%

```

```

        \glsinsert}%
    }%
}%
{%
    \glscapscase
    {%
        \acronymfont
        {\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \acronymfont
        {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{%
            \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
            \glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \mfirstucMakeUppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
}%

```

```

    }%
  }%
}%
{%
  \glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay
{\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax

```

```

\let\glsifplural\@secondoftwo
\let\glsupcase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{%
    \MakeUppercase{\glsentrylong{#2}}}{#2}#3}%
  }%
  \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}{#2}#3}%
}
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\Glsentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@text@accessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\glsentryfirst{#2}}{#2}#3}%
}
\def\@Gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@first@accessdisplay{\mfirstucMakeUppercase{\glsentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\glsentryplural{#2}}{#2}#3}%
}
\def\@Gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\Glsentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@plural@accessdisplay{\mfirstucMakeUppercase{\glsentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}
\def\@Gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

```

```

\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glsentryfirstplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbolplural@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}}%
}
\def\@glsuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glentryuseri{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}}%
}
\def\@glsuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glentryuserii{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}}%
}
\def\@glsuseriii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glentryuseriii{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}}%
}
\def\@glsuseriv@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%

```



```

\@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

```

```

\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}
\newcommand*{\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*{\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
}

```

```

\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%

```

```

\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{sc-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}

```

```

}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
      {%
        \glscapscase
        {%
          \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
        }%
        {%
          \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
        }%
        {%
          \glslongaccessdisplay
            {\mfirstucMakeUppercase
              {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \glsgenentryfmt
  }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
      {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%

```

```

\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}

```

```

}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshort{##2}}}{##2}##3\space
        (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}}%
\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\gslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\gslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\gslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\gslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%

```



```

}
\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\newcommand*{\glsdefaultshortaccess}[2]{#1}
\renewcommand*{\newacronymhook}{%
  \protected@edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,%
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,%
      firstpluralaccess=\relax,%
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
}
\let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\gl@labeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\gl@shorttok}},%
      sort={\the\gl@shorttok},%
      text={\the\gl@shorttok},%
      short={\the\gl@shorttok},%
      shortplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
      shortaccess={\gl@defaultshortaccess{\the\gl@longtok}{\the\gl@shorttok}},%
      long={\the\gl@longtok},%
      longplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
      symbol={\the\gl@longtok},%
      symbolplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\gl@keylisttok
    }%
  }%
\let\org@gl@assign@firstpl\gl@assign@firstpl
\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%

```

```

}%
\@do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}

```

```

\renewcommand*{\FootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      textaccess={\the\glslongtok},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@descplural\@org@gls@assign@descplural
}
\renewcommand*{\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      access={\noexpand\@glo@symbolaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

text={\noexpand\@glo@short},%
textaccess={\noexpand\@glo@shortaccess},%
plural={\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\@glo@firstplural},%
symbol={\the\glsshorttok},%
symbolaccess={\the\glslongtok},%
symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglonameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglopluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglofirstpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglosymbolaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglosymbolpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}
\newcommand*{\showglodescaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*{\showglodescpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*{\showgloshortaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*{\showgloshortpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*{\showglolongaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*{\showglolongpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}

```

8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak

```

```

        Language modules need to be installed separately.\MessageBreak
        Please check on CTAN for a bundle called\MessageBreak
        ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
}%
}%
{}%

```

8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2021/11/01 v4.49 (NLCT)]
\ifglxsindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J}%
    (markup-locref
    :open \string"\string~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{glshyphnumber}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshyphnumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}

```



```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxsindy
  \renewcommand*{\GlsAddXdyLocation}[2]{%
    \edef\xdyuserlocationdefs{%
      \xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\xdyuserlocationnames{%
      \xdyuserlocationnames^^J\space\space\space
      \string"#1\string"%
    }
  }
\fi
\renewcommand{\@do@wrglossary}[1]{%
\ifglxsindy
  \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \protected@edef\@glslocref{\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=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
  \@onelevel@sanitize\@gls@roman
  \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}%
  \@onelevel@sanitize\@tmp
  \ifx\@tmp\@gls@roman
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (\string"roman-numbers-lowercase\string")
      :min-range-length \@glsminrange)}}%
  \else
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (:sep "\@gls@roman")
      :min-range-length \@glsminrange)}}%
  \fi
  \write\glswrite{(define-location-class
    \string"Roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-uppercase\string")
    :min-range-length \@glsminrange)}}%
  \write\glswrite{(define-location-class
    \string"arabic-page-numbers\string"^^J\space\space\space
    (\string"arabic-numbers\string")
    :min-range-length \@glsminrange)}}%
  \write\glswrite{(define-location-class
    \string"alpha-page-numbers\string"^^J\space\space\space
    (\string"alpha\string")
    :min-range-length \@glsminrange)}}%

```

```

\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
    :min-range-length \@glsmminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
    :sep \string"\@glAlphacompositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
    :sep \string"\glscpositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{@xduyuserlocationdefs}%
\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\glseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xduylocationclassorder))}%
\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\glresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsoopenbrace\string\glossaryentrynumbers
  \glsoopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list

```

```

:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\string~\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"glslnumbers\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]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
    \csuse{@glscmpstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
    \csuse{@glscmpstyle@list}%
}%
\compatglossarystyle{altlist}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[\glentryitem{##1}\glstarget{##1}{##2}]{%
            \mbox{}\par\nobreak\@afterheading
            ##3\glspostdescription\space ##5}%
    \renewcommand{\glossarysubentryfield}[6]{%
        \par
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
    \csuse{@glscmpstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
    \csuse{@glscmpstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[]\makebox[\glslistdottedwidth][l]{%
            \glentryitem{##1}\glstarget{##1}{##2}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
    \renewcommand*{\glossarysubentryfield}[6]{%
        \item[]\makebox[\glslistdottedwidth][l]{%
            \glssubentryitem{##2}%
            \glstarget{##2}{##3}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
}%
\compatglossarystyle{sublistdotted}{%
    \csuse{@glscmpstyle@listdotted}%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[\glentryitem{##1}\glstarget{##1}{##2}]{%
}%
\compatglossarystyle{long}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
    \renewcommand*{\glossarysubentryfield}[6]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}

```

```

}%
\compatglossarystyle{longborder}{%
  \csuse{@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{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##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]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%

```

```

\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glstarget{##1}{\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
      \else
        \space(##4)%
      \fi
    \space ##3\glspostdescription \space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \ifcase##1\relax
      % level 0
      \item
    \or
      % level 1
      \subitem
    \glssubentryitem{##2}%
  \else
    % all other levels
    \subsubitem
  \fi
  \textbf{\glstarget{##2}{##3}}%
  \ifx\relax##5\relax
    \else
      \space(##5)%
    \fi
  \space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstarget{##1}{\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
      \else
        \space(##4)%
      \fi
    \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}%
}%

```

```

\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{alttree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else

```

```

\settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[0pt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription\space ##5\par
\def\@gls@prevlevel{0}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\ifnum\@gls@prevlevel=##1\relax
\else
\@ifundefined{\@glswidestname\romannumeral##1}{%
\settowidth{\@gls@tmplen}{\textbf{\@glswidestname\space}}{%
\settowidth{\@gls@tmplen}{\textbf{%
\csname \@glswidestname\romannumeral##1\endcsname\space}}}%
\ifnum\@gls@prevlevel<##1\relax
\setlength\glstreeindent\@gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{\@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\textbf{%
\@glswidestname\space}}}%
\settowidth{\glstreeindent}{\textbf{%
\csname \@glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[0pt][r]{\makebox[\@gls@tmplen][l]{%
\textbf{\glstarget{##2}{##3}}}%
\ifx##5\relax\relax
\else
(##5)\space
\fi
##4\glspostdescription\space ##6\par
\def\@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreeegroup}{%

```

```

\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreename}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%

```

```

\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \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]{%
\ifcsemtty{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
\ifcsemtty{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}{\glentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
            {\glentryprefixfirstplural{#2}\glsprefixsep}{}%
        }%
        \@glsp1@{#1}{#2}[#3]%
    }%
}
\newrobustcmd{\Pgls}{\@glshyp@opt\Pgls}
\newcommand*{\@Pgls}[2][{}]{%
    \new@ifnextchar[%
        {\@Pgls@{#1}{#2}}%
        {\@Pgls@{#1}{#2}[]}%
    }
\def\Pgls@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glstoifexists{#2}%
    {%
        \ifglused{#2}%
        {%
            \ifglshasprefix{#2}%
            {%
                \Glentryprefix{#2}%
                \glsprefixsep
                \@glsh@{#1}{#2}[#3]%
            }%
            {\@Glsh@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {%
                \Glentryprefixfirst{#2}%
                \glsprefixsep
                \@glsh@{#1}{#2}[#3]%
            }%
            {\@Glsh@{#1}{#2}[#3]}%
        }%
    }%
}
\newrobustcmd{\Pglspl}{\@glshyp@opt\Pglspl}
\newcommand*{\@Pglspl}[2][{}]{%
    \new@ifnextchar[%
        {\@Pglspl@{#1}{#2}}%
        {\@Pglspl@{#1}{#2}[]}%
    }
\def\Pglspl@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glstoifexists{#2}%
    {%

```

```

\ifglused{#2}%
{%
  \ifglshasprefixplural{#2}%
  {%
    \Glstryentryprefixplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#1}{#2}[#3]}%
}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glstryentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glsp1@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@glshyp@opt\PGLS}
\newcommand*{\@PGLS}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2}[]}%
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstdoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glstryentryprefix{#2}\glsprefixsep}}}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glstryentryprefixfirst{#2}\glsprefixsep}}}%
    }%
    \@GLS@{#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}%
\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \ifglschasprefixplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{}%
  }%
  {%
    \ifglschasprefixfirstplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}%
  }%
  \@GLSpl@{#1}{#2}[#3]%
}%
}

```

8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \@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@rerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \@ifundefined{\@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
  }{%

```

```

\expandafter\let\expandafter\@gls@tmp
\csname @gls@hypergroupplist@#1\endcsname
\expandafter\xdef\csname @gls@hypergroupplist@#1\endcsname{%
\@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
\def\@gls@between{}%
\ifcsundef\@gls@hypergroupplist@\@glo@type}%
{%
\def\@gls@list{}%
}%
{%
\expandafter\let\expandafter\@gls@list
\csname @gls@hypergroupplist@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
\@gls@between
\@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
\glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
\let\@gls@between\glshypernavsep
}%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glssymbolnav}{%
\glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
\glshypernavsep
\glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
\glshypernavsep
}

```

8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2021/11/01 v4.49 (NLCT)]
\newglossarystyle{inline}{%
\renewenvironment{theglossary}%
{%
\def\gls@inlinesep{}%
\def\gls@inlinesubsep{}%
\def\gls@inlinepostchild{}%
}%
{\glspostinline}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsinlinedopostchild
\gls@inlinesep
\glsentryitem{##1}%

```

```

\glsinlinenameformat{##1}{%
  \glossentryname{##1}%
}%
\ifglshasdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\ifglshaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{%
  }%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glsesubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}

```

```

\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.27 Rollback v4.49 (glossary-list-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-list}[2021/11/01 v4.49 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glslistgroupheaderfmt}[1]{#1}
\newcommand*{\glslistnavigationitem}[1]{\item[#1]}
\newcommand*{\glslistinit}{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
}%
\newcommand{\glslistexpandedname}[1]{%
  \glsunexpandedfieldvalue{#1}{name}%
}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}{%
    {\glslistinit\begin{description}}{\end{description}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}]
    \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*{\subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
}

```



```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
    \renewcommand{\subglossentry}[3]{%
      \par
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
    }
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*{\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
    \renewcommand*{\subglossentry}[3]{%
      \item[\makebox[\glslistdottedwidth][l]{%
        \glssubentryitem{##2}%
        \glstarget{##2}{\glossentryname{##2}}%
        \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
      }
    }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%

```

```

\item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}}%
}

```

8.28 Rollback v4.49 (glossary-long-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}[l|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{\glstdescwidth}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l|l|p{\glstdescwidth}|p{\glspagelistwidth}|}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
  }
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
  }

```

```

}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

```

\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
    {\end{longtable}}%
}
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%

```

```

        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
\ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
\else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long4col}%
    \renewcommand*{\glossaryheader}{%
        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \symbolname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
    \fi
}
\newglossarystyle{altlong4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long4col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long3col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}%
            >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%

```

```

\glspatchLToutput
\setglossarystyle{altlong4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@ \pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
            \@outputpage
            \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
          \fi
          \fi
          \global\@colroom\@colht
          \global\vsizel\@colht
          {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
      \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
      \fi
    }%
  }
}

```

8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l>{\raggedright}p{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}%
    \glspostdescription\space ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{glsdescwidth}|}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}
```



```

}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[ll>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline

```

```

\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
\renewenvironment{theglossary}%
{\begin{longtable}[l>\raggedright]p{\glsdescwidth}l%
>\raggedright}p{\glspagelistwidth}}}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgruppeheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} & \glossentrysymbol{##1} &
##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\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|l>\raggedright]p{\glsdescwidth}|l|%
>\raggedright}p{\glspagelistwidth}|}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
\setglossarystyle{altlongragged4col}%
\renewenvironment{theglossary}%

```

```

        {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}%
          >{\raggedright}p{\glspagelistwidth}|}}%
        {\end{longtable}}}%
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}

```

8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.49]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}{%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}}%
}
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsggetgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsggetgrouptitle{##1}}}%
    \indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%
}

```

```

\renewenvironment{theglossary}%
{
  \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
  \setlength{\parindent}{0pt}%
  \setlength{\parskip}{0pt plus 0.3pt}%
  \let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}

```

```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalttreegroup}{%
  \setglossarystyle{mcolalttree}%

```

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalittreehypergroup}{%
\setglossarystyle{mcolalittree}%
\renewcommand*\{glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\{glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcolalittreespannav}{%
\setglossarystyle{alittree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*\{glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
\newlength{glsdescwidth}
\setlength{glsdescwidth}{0.6\hspace}
}{}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsgnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|lp{\glstdescwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
    \bfseries\descriptionname\tabularnewline}%
    \tabletail{}%
    {\begin{supertabular}{lp{\glstdescwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
    \tabletail{\hline}
    {\begin{supertabular}{|lp{\glstdescwidth}|}%

```

```

        {\end{supertabular}}}%
    }
\newglossarystyle{super3col}{%
    \renewenvironment{theglossary}%
        {\tablehead{}\tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
    \fi
}
\newglossarystyle{super3colborder}{%
    \setglossarystyle{super3col}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
        {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
        {\end{supertabular}}}%
}
\newglossarystyle{super3colheader}{%
    \setglossarystyle{super3col}%
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
            \bfseries\pagelistname\tabularnewline}\tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}}%
}
\newglossarystyle{super3colheaderborder}{%
    \setglossarystyle{super3colborder}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline
            \bfseries\entryname&\bfseries\descriptionname&
            \bfseries\pagelistname\tabularnewline\hline}%
        {\tabletail{\hline}%
        {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
        {\end{supertabular}}}%
}

```



```

\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{l|l|l|l}}{%
      \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    {\tabletail{}}%
    {\begin{supertabular}{l|l|l|l}}{%
      \end{supertabular}}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|l|l|l|l|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super4colheaderborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    {\tabletail{\hline}%
    {\begin{supertabular}{|l|l|l|l|}}%
    {\end{supertabular}}}%
}

```

```

}
\newglossarystyle{altsuper4col}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}%
      {\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
  \setglossarystyle{super4colheaderborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    {\begin{supertabular}%
      {\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}}%
    {\end{supertabular}}}%
}

```

8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsgdescwidth}{%
  \newlength{glsgdescwidth}
  \setlength{glsgdescwidth}{0.6\hspace}
}{%
}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{||l>{\raggedright}p{\glstdescwidth}||}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glstdescwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%

```

```

\tabletail{\hline}
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||}%
{\end{supertabular}}}%
}
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
{\tablehead{} \tabletail{}}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
##3\tabularnewline
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
>{\raggedright}p{\glspagelistwidth}||}%
{\end{supertabular}}}%
}
\newglossarystyle{superragged3colheader}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline}\tabletail{}}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}}%
}
\newglossarystyle{superragged3colheaderborder}{%
\setglossarystyle{superragged3colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|l|>{\raggedright}p{\glsgdescwidth}|%
        >{\raggedright}p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
        \begin{supertabular}{|l>{\raggedright}p{\glsgdescwidth}l%
            >{\raggedright}p{\glspagelistwidth}|}%
        {\end{supertabular}}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} &
        \glossentrysymbol{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
    \fi
}
\newglossarystyle{altsuperragged4colheader}{%
    \setglossarystyle{altsuperragged4col}%
    \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline}\tabletail{}%
        \begin{supertabular}{|l>{\raggedright}p{\glsgdescwidth}l%
            >{\raggedright}p{\glspagelistwidth}|}%
        {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
    \setglossarystyle{altsuper4col}%
    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
        \begin{supertabular}%
            { |l|>{\raggedright}p{\glsgdescwidth}|l|%
                >{\raggedright}p{\glspagelistwidth}|}%
        {\end{supertabular}}}%
}

```

```

}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {||>{\raggedright}p{\glsdescwidth}||%
        >{\raggedright}p{\glspagelistwidth}||}%
    {\end{supertabular}}%
}

```

8.34 Rollback v4.49 (glossary-tree-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2021/11/01 v4.49 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%

```

```

\item\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{tree}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}{%
    {\setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}}%
  }{%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi

```



```

\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glslshhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glswidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\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@glsetoklabel{\@glo@label}@name}%
\fi
}%
}%
}%
}
\newcommand*{\glswidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][l]{#2}%
}
\newglossarystyle{alttree}{%

```

```

\renewenvironment{theglossary}%
{
  \def\@gls@prevlevel{-1}%
  \mbox{}\par}%
{\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
  \hangindent\glstreeindent
  \parindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}{%
            \settowidth{\glstreeindent}{\glstreenamefmt{%
              \csname @glswidestname\romannumeral\@gls@prevlevel
                \endcsname\space}}}%
          \addtolength\parindent{-\glstreeindent}%
          \setlength\glstreeindent\parindent
        \fi
        \fi
      \hangindent\glstreeindent
      \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
        \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
      \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
      \glossentrydesc{##2}\glspostdescription\space ##3\par
      \def\@gls@prevlevel{##1}%
    }%
  \fi
}

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

Change History

1.01 – 2007-05-17	
General: Added range facility in format key	124
\writeist: Added spaces after \delimN and \delimR in ist file	178
1.04 – 2007-08-03	
General: Added \glstextformat	112
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 \theglentrycounter setting the page number too soon	122
\glsadd: fixed bug caused by \theglentrycounter setting the page number too soon	175
1.08 – 2007-10-13	
General: Added babel support	45
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle	276
listgroup: changed listgroup style to use \glsgetgrouptitle	275
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	127
\@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)	133
\@GLSpl: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	139
\@GLspl0: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	138
\@glspl0: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	137
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	167
\glentrydescplural: New	167
\Glsentrysymbolplural: New	168
\glentrysymbolplural: New	168
symbolplural: new	77
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	141–149
\ACRfullpl: new	243
\Acrfullpl: new	243
\acrfullpl: new	242
\acrpluralsuffix: New	240
\gls@defglossaryentry:	
Changed default first value	95

Changed default firstplural value	95	checking if footnote option has been used	138
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	135
<code>\newacronym</code> : Removed restriction on only using <code>\newacronym</code> in the preamble	239	<code>\@glsdisp</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	140
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	137
General: added nonumberlist key to <code>\printglossary</code>	226	<code>\@gls@target</code> : raised the hypertarget so the target text doesn't scroll off the top of the page	133
added numberedsection key to <code>\printglossary</code>	225	<code>\gls@defglossaryentry</code> : Changed def to let	95
<code>\@gls@hypergroup</code> : new	270	1.17 – 2008-12-26	
<code>\firstacronymfont</code> : new	243	General: added xindy support	32
<code>\glsautoprefix</code> : new	10	<code>\@do@esc@wrglossary</code> : new	204
<code>\glsnavhyperlink</code> : changed <code>\edef</code> to <code>\protected@edef</code>	268	<code>\@do@seeglossary</code> : new	208
<code>\glsnavhypertarget</code> : added write to aux file	269	<code>\@glo@storeentry</code> : new	102
<code>\glsnavigation</code> : changed to only use labels for groups that are present	270	<code>\@gls@glossary</code> : changed definition to use <code>\index</code> instead of <code>\@index</code>	200
1.15 – 2008-08-15		<code>\@glsdefaultplural</code> : new	80
<code>\@gls@link</code> : added <code>\glslabel</code>	122	<code>\@glsdefaultsort</code> : new	81
<code>\gls@defglossaryentry</code> : check for <code>\@glo@first</code> in description	99	<code>\@gls@hypernumber</code> : new	236
check for <code>\@glo@text</code> in symbol	99	<code>\@gls@noname</code> : new	80
<code>\gls@hypergroup@rerun</code> : new	270	<code>\@gls@nonextpages</code> : new	227
<code>\glsnavhypertarget</code> : added check if rerun required	269	<code>\gls@defglossaryentry</code> : added nonumberlist key	96
<code>\glssettoctitle</code> : new	44	added parent key	95
<code>\printglossary</code> : changed the way the TOC title is set	211	added see key	95
1.16 – 2008-08-27		Stored main part of entry format when entry is defined	100
<code>\@GLS@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	136	<code>\gls@suffixF</code> : new	49
<code>\@GLSpl</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	139	<code>\gls@suffixFF</code> : new	49
<code>\@Gls@</code> : Test glossary type is <code>\acronymtype</code> in addition to checking if footnote option has been used	136	<code>\gls@wrglossary</code> : modified to allow for xindy support	200
<code>\@Glspl@</code> : Test glossary type is <code>\acronymtype</code> in addition to		<code>\gls@hyperlink</code> : new	174
		<code>\gls@hypernumber</code> : modified to allow material to be attached to location	235
		<code>\glsnavhyperlink</code> : replaced <code>\hyperlink</code> to <code>\@glslink</code>	268

\glsnavhypertarget: replaced	\@printglossary: changed \xdef
\hypertarget to	to \protected\xdef 213
\@gls@target 269	\gls@defglossaryentry: changed
\glssee: new 209	\xdef to
\glsseeformat: new 209	\protected\xdef 97, 100
\glsSetSuffixF: new 49	\glsaddprotectedpagefmt:
\glsSetSuffixFF: new 49	changed \eappto to
\ifglxindy: new 32	\protected@eappto 202
\istfilename: added xindy	2.01 – 2009 May 30
support 48	\@gls@link: moved
\newglossarystyle: made	\@do@wrglossary before term
\newglossarystyle long . . 234	is displayed to prevent
nonumberlist: new 78	unwanted whatsit 123
\nopostdesc: new 47	\forallglossaries: replaced
order: order package option	\ifthenelse with \ifx 63
added 32	\forallglsentries: replaced
parent: new 78	\ifthenelse with \ifx 64
\printglossary: added check to	\glsdefmain: new 19
determine if \printglossary	\glsdescwidth: changed
is already defined 211	\linewidth to \hsize . 278, 301
added print language to aux	\glslistdottedwidth: changed
file 211	\linewidth to \hsize 277
see: new 77	\glspagelistwidth: changed
\writeist: added xindy support 178	\linewidth to \hsize . 278, 301
1.18 – 2009-01-14	nomain: added nomain package
\@gls@loadlist: new 13	option 20
\@gls@loadlong: new 12	\writeist: removed item_02 -
\@gls@loadsuper: new 12	no such makeindex key . . . 183
\@gls@loadtree: new 13	2.02 – 2007-07-13
\gls@defglossaryentry:	\@printglossary: suppressed
Changed default value of sort	warning globally rather than
to \glsdefaultsort 95	locally 214
moved sort sanitization to	2.02 – 2009-07-13
\newglossaryentry 99	\glossarysection: changed
\glstarget: new 230	\@mkboth to \glossarymark 51
nolist: new 13	\gls@glossarymark: New 51
nolong: new 12	2.03 – 2009-09-23
nostyles: new 13	\@GLS@: Added check for
nosuper: new 13	hyperfirst 136
notree: new 13	\@GLSpl: Added check for
\oldacronym: new 238	hyperfirst 139
sort: moved sanitization to	\@GLS@: Added check for
\newglossaryentry 76	hyperfirst 136
1.19 – 2009-03-02	\@GLspl@: Added check for
\gls@clearpage: new 54	hyperfirst 138
\glsdisp: new 139	\@gls@: Added check for
1.46 – 2021-09-20	hyperfirst 135
General: changed \edef to	\@gls@link: new 120
\protected@edef 369	\@gls@link: added \leavevmode 122

Moved entry existence check to avoid duplicate code	122	\glsadd: fixed bug that ignored counter	175
\@glsdisp: Added check for hyperfirst	140	\Glsentryuseri: new	170
\@glspl@: Added check for hyperfirst	137	\glsentryuseri: new	170
\gls glossarymark: Added check to see if it's already defined .	51	\Glsentryuserii: new	170
hyperfirst: new	31	\glsentryuserii: new	170
2.04 – 2009-11-10		\Glsentryuseriii: new	170
\@GLS@: Changed test to check if glossary type has been identified as a list of acronyms	136	\glsentryuseriii: new	170
\@GLSpl: Changed test to check if glossary type has been identified as a list of acronyms	139	\Glsentryuseriv: new	171
\@Gls@: Changed test to check if glossary type has been identified as a list of acronyms	136	\glsentryuseriv: new	171
\@Glspl@: Changed test to check if glossary type has been identified as a list of acronyms	138	\Glsentryuserv: new	171
\@glossaryentryfield: new . .	101	\glsentryuserv: new	171
\@glossarysubentryfield: new	101	\Glsentryuservi: new	171
\@gls@: Changed test to check if glossary type has been identified as a list of acronyms	135	\glsentryuservi: new	171
\@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	140	\SetAcronymLists: new	22
\@glspl@: Changed test to check if glossary type has been identified as a list of acronyms	137	\SetDefaultAcronymDisplayStyle: new	258
\@newglossaryentryposthook: new	101	\SetDefaultAcronymStyle: new	259
\@newglossaryentryprehook: new	101	\SetDescriptionAcronymDisplayStyle: new	260
acronymlists: new	22	\SetDescriptionDUAAcronymDisplayStyle: new	259
\DeclareAcronymList: new . . .	21	\SetDescriptionFootnoteAcronymDisplayStyle: new	259
\DefineAcronymSynonyms: new .	257	\SetDUADisplayStyle: new . . .	261
\gls@defglossaryentry: added user1-6 keys	96	\SetFootnoteAcronymDisplayStyle: new	260
		\SetSmallAcronymDisplayStyle: new	260
		2.05 – 2010-02-06	
		\@glsdisp: Added closing brace. Patch provided by Sergiu Dotenco	139
		Removed spurious brace. Patch provided by Sergiu Dotenco	140
		\writeist: Added \string before opening and closing braces. Patch provided by Sergiu Dotenco	183
		2.06 – 2010-06-14	
		\altnewglossary: new	73
		\CustomAcronymFields: new . .	261
		\CustomNewAcronymDef: new . .	261
		\SetCustomDisplayStyle: new .	261
		\SetCustomStyle: new	261

2.07 – 2010-07-10		\@xdyattributelist: new	54
General: glsadd format key stored		\acrfootnote: new	259
in \@glsnumberformat (was		\ACRfull: added starred version	242
mistakenly stored in		\Acrfull: added starred version	242
\@glo@format)	175	\acrfull: added starred version	242
3.0 – 2010-07-12		\ACRfullpl: added starred	
\@makeglossary: Added check for		version	243
savewrites	188	\Acrfullpl: added starred	
\@gls@wrglossary: modified to		version	243
take into account savewrites	200	\acrfullpl: added starred	
3.0 – 2010/03/31		version	243
\@set@glo@numformat: added 4th		\acrlinkfootnote: new	259
argument	124	\acrnolinkfootnote: new	259
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	38
hyperlink	237	counter: replaced \@ifundefined	
etoolbox now loaded	6	with \ifcsundef	77
replaced \@ifundefined with		counterwithin: new	14
\ifcsundef	43, 46, 118, 225	entrycounter: new	14
\@do@esc@wrglossary: added		glossaryentry: new	14
check for hyper location		\glossarysection: replaced	
prefix	206	\@ifundefined with	
modified to use new format .	204	\ifcsundef	51
\@@glossarysec: replaced		\glossarystyle: replaced	
\@ifundefined with		\@ifundefined with	
\ifcsundef	10	\ifcsundef	234
\@do@seeglossary: Sanitize and		glossarysubentry: new	15
escape cross-referencing		\gls@codepage: replaced	
information	208	\@ifundefined with	
\@gls@counterwithin: new . . .	15	\ifcsundef	33
\@gls@ifinlist: new	55	\gls@defglossaryentry: added	
\@gls@link: added		\@gls@defsort	99
\@gls@saveentrycounter .	123	added short and long keys .	96
added \@gls@setsort	123	replaced \@ifundefined with	
\@gls@saveentrycounter: new .	123	\ifcsundef	96
\@gls@setupsort@def: new . . .	17	\gls@doclearpage: replaced	
\@gls@setupsort@standard: new	16	\@ifundefined with	
\@gls@setupsort@use: new . . .	17	\ifcsundef	53
\@gls@xdy@locationlist: new .	58	\glsadd: added	
\@glslink: replaced		\@gls@saveentrycounter .	175
\@ifundefined with		\GlsAddXdyCounters: new	55
\ifcsundef	133	\glentrycounterlabel: new .	228
\@glsnextpages: new	227	\glentryitem: new	229
\@print@glossary: replaced		\Glentrylong: new	172
\@ifundefined with		\glentrylong: new	172
\ifcsundef	214	\Glentrylongpl: new	172
\@printglossary: added		\glentrylongpl: new	172
\currentglossary	213	\Glentryshort: new	172
added \glsnextpages	213	\glentryshort: new	171
make toctitle default to title	213	\Glentryshortpl: new	172

<code>\glstryshortpl</code> : new	172	<code>\oldacronym</code> : replaced	
<code>\glsgtgroupitl</code> : replaced		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	233	<code>\ifcsundef</code>	238
<code>\glsglossarymark</code> : replaced		<code>\printglossary</code> : replaced	
<code>\@ifundefined</code> with		<code>\@ifundefined</code> with	
<code>\ifcsundef</code>	51	<code>\ifcsundef</code>	211
<code>\glshyperlink</code> : changed default		<code>savewrites</code> : new	38
from <code>\glstryname</code> to		<code>see</code> : added	
<code>\glstrytext</code>	174	<code>\@glo@seeautonumberlist</code>	77
<code>\glshypernumber</code> : replaced		<code>seeautonumberlist</code> : new	12
<code>\@ifundefined</code> with		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\ifcsundef</code>	235	expanded options link	
<code>\glsnrformat</code> : replaced		options	259
<code>\@ifundefined</code> with		<code>\setentrycounter</code> : added	
<code>\ifcsundef</code>	50	optional argument	234
<code>\glstryentry</code> : new	228	<code>short</code> : new	80
<code>\glstryresetentrycounter</code> :		<code>shortplural</code> : new	80
new	228	<code>\showacronymlists</code> : new	266
<code>\glstryseeitem</code> : hyperlink uses		<code>\showgloccounter</code> : new	263
<code>\glstryseeitemformat</code> instead of		<code>\showgloccounter</code> : new	265
<code>\glstryname</code>	210	<code>\showgloccounterpl</code> : new	265
<code>\glstryseeitemformat</code> : new	210	<code>\showgloccounterfirst</code> : new	263
<code>\glssortnumberfmt</code> : new	16	<code>\showgloccounterfirstpl</code> : new	263
<code>\glstrystepentry</code> : new	228	<code>\showgloccounterflag</code> : new	266
<code>\glstrystepsubentry</code> : new	228	<code>\showgloccounterindex</code> : new	266
<code>\glstrysubentrycounterlabel</code> :		<code>\showgloccounterlevel</code> : new	262
new	229	<code>\showgloccountername</code> : new	264
<code>\glstrysubentryitem</code> : new	229	<code>\showgloccounterparent</code> : new	262
<code>\ifglossaryexists</code> : replaced		<code>\showgloccounterplural</code> : new	263
<code>\@ifundefined</code> with		<code>\showgloccountersort</code> : new	265
<code>\ifcsundef</code>	64	<code>\showglossaries</code> : new	267
<code>\ifglstryentryexists</code> : replaced		<code>\showglossarycounter</code> : new	267
<code>\@ifundefined</code> with		<code>\showglossaryentries</code> : new	267
<code>\ifcsundef</code>	65	<code>\showglossaryin</code> : new	267
<code>\istfile</code> : deprecated	198	<code>\showglossaryout</code> : new	267
<code>long</code> : new	80	<code>\showglossarytitle</code> : new	267
<code>longplural</code> : new	80	<code>\showglossymbol</code> : new	265
<code>\newglossaryentry</code> : replaced		<code>\showglossymbolplural</code> : new	265
<code>\DeclareRobustCommand</code> with		<code>\showglostext</code> : new	262
<code>\newrobustcmd</code>	83	<code>\showglostype</code> : new	263
<code>\newglossarystyle</code> : replaced		<code>\showglouseri</code> : new	263
<code>\@ifundefined</code> with		<code>\showglouserii</code> : new	264
<code>\ifcsundef</code>	234	<code>\showglouseriii</code> : new	264
<code>nonumberlist</code> : now boolean	78	<code>\showglouseriv</code> : new	264
<code>\ns@newglossary</code> : added		<code>\showglouseriv</code> : new	264
<code>\@gls@defsortcount</code>	73	<code>\showglouseriv</code> : new	264
replaced <code>\@ifundefined</code> with		<code>sort</code> : new	15
<code>\ifcsundef</code>	73	<code>subentrycounter</code> : new	15

theglossary: replaced	\glsname: made robust	144
\@ifundefined with	\GLSpl: made robust	138
\ifcsundef	\Glspl: made robust	137
\writeist: added xindy-only	\glspl: made robust	137
macro definitions to glossary	\GLSplplural: made robust	143
open tag	\GLSsymbol: made robust	148
modified to support new	\Glsymbol: made robust	148
format	\glsymbol: made robust	147
3.01 – 2011-04-12	\GLSsymbolplural: made robust	149
General: made robust	\Glsymbolplural: made robust	149
\@glswritefiles: added check	\glsymbolplural: made robust	148
for empty glossaries	\GLstext: made robust	141
\ACRfull: made robust	\glstext: made robust	140
\Acrfull: made robust	\GLSuseri: made robust	150
\acrfull: made robust	\Glsuseri: made robust	150
\acrfullformat: removed	\glsuseri: made robust	149
\acronymfont as it should	\GLSuserii: made robust	151
already be set in the second	\Glsuserii: made robust	151
argument.	\glsuserii: made robust	150
\ACRfullpl: made robust	\GLSuseriii: made robust	152
\Acrfullpl: made robust	\Glsuseriii: made robust	152
\acrfullpl: made robust	\glsuseriii: made robust	151
\ACRlong: made robust	\GLSuseriv: made robust	153
\Acrlong: made robust	\Glsuseriv: made robust	153
\acrlong: made robust	\glsuseriv: made robust	152
\ACRlongpl: made robust	\GLSuserv: made robust	154
\Acrlongpl: made robust	\Glsuserv: made robust	154
\acrlongpl: made robust	\glsuserv: made robust	153
\ACRshort: made robust	\GLSuservi: made robust	155
\Acrshort: made robust	\Glsuservi: made robust	155
\acrshort: made robust	\glsuservi: made robust	154
\ACRshortpl: made robust	3.02 – 2012-05-19	
\Acrshortpl: made robust	\glsnumlistlastsep: new	174
\acrshortpl: made robust	\glsnumlistsep: new	174
\Gls: made robust	3.02 – 2012-05-21	
\glsadd: made robust	\@do@wrglossary: changed	
\glsaddall: made robust	\@glslocref to	
\GLSdesc: made robust	\theglsentrycounter	207
\Glsdesc: made robust	\@do@wrglossary: changed	
\glsdesc: made robust	\@do@wrglossary to test for	
\GLSdescplural: made robust	indexonlyfirst option; put old	
\Glsdescplural: made robust	\@do@wrglossary code into	
\glsdescplural: made robust	\@do@wrglossary	201
\glsfirst: made robust	\@gls@missingnumberlist: new	81
\GLSfirstplural: made robust	\@glswritefiles: added check	
\Glsfirstplural: made robust	for existence of token in case	
\glsfirstplural: made robust	\makeglossaries has been	
\glslink: made robust	omitted	198
\GLSname: made robust	\@printglossary: add a way to	
\Glsname: made robust	fetch current entry label	213

<code>\gls@defglossaryentry</code> : added numberlist element	99	<code>\gls@postinline</code> : replaced “.” with <code>\gls@postdescription</code>	273
<code>\gls@save@numberlist</code> : new	210	<code>index</code> : added check for glsnogroupskip	316
<code>\gls@wrglossary</code> : added check for glossary file defined	200	<code>list</code> : added check for glsnogroupskip	275
<code>\glsdisplaynumberlist</code> : new	173	<code>long</code> : added check for glsnogroupskip	279
<code>\glsentrycounter</code> : set default value	123	<code>long3col</code> : added check for glsnogroupskip	280
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	173	<code>long4col</code> : added check for glsnogroupskip	282
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>)	173	<code>longragged</code> : added check for glsnogroupskip	290
<code>\glsentrynumberlist</code> : new	173	<code>longragged3col</code> : added check for glsnogroupskip	291
<code>\glsmoveentry</code> : new	101	<code>nogroupskip</code> : new	14
<code>\glsresetentrycounter</code> : new	228	<code>nopostdot</code> : new	13
<code>\ifglshaschildren</code> : new	67	<code>super</code> : added check for glsnogroupskip	301
<code>\ifglshasparent</code> : new	67	<code>super3col</code> : added check for glsnogroupskip	303
<code>indexonlyfirst</code> : new	31	<code>super4col</code> : added check for glsnogroupskip	305
<code>\makeglossaries</code> : added list parser	193	<code>superragged</code> : added check for glsnogroupskip	309
<code>\renewglossarystyle</code> : new	235	<code>superragged3col</code> : added check for glsnogroupskip	311
<code>savenumberlist</code> : new	12	<code>tree</code> : added check for glsnogroupskip	318
<code>\showglossaryentries</code> : fixed misspelt command	267	<code>treenoname</code> : added check for glsnogroupskip	319
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	260		
<code>ucmark</code> : new	14		
3.03 – 2012/09/21		3.04 – 2012-11-11	
General: allow title to set toctitle	225	<code>altlist</code> : replaced <code>\newline</code> with paragraph break	276
<code>\@gls@sanitizesort</code> : new	25	3.04 – 2012-11-18	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	16	General: Added check for doc package	6
<code>\@printglossary</code> : allow title to override default toctitle	212	added datatool-base as a required package	6
<code>altlongragged4col</code> : added check for glsnogroupskip	293	added local key	119
<code>altsuperragged4col</code> : added check for glsnogroupskip	312	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	207
<code>alttree</code> : added check for glsnogroupskip	322	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	205
<code>\glsinlinedescformat</code> : new	273	<code>\@gls@esc@csdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphapage</code> ,	
<code>\glsinlineemptydescformat</code> : new	273		
<code>\glsinlinenameformat</code> : new	273		
<code>\glsinlinepostchild</code> : new	273		
<code>\glsinlinesubdescformat</code> : new	273		
<code>\glsinlinesubnameformat</code> : new	273		

\gls@Alphpage and \gls@romanpage	125	\findrootlanguage: Obsoleted	62
\@print@glossary: Moved aux write to end of document to prevent unwanted whatsit occurring here.	214	3.07 – 2013-07-05 \@gls@link: fixed bug that failed to find entry in list	122
\gls@Alphpage: new	202	\glossarypreamble: modified to work with \setglossarypreamble	50
\gls@alphpage: new	201	\gls@docclearpage: added check for openright	53
\gls@disablepagerefexpansion: new	201	\glspostdescription: Added spacefactor code	13
\gls@numberpage: new	202	\GlsSetXdyCodePage: Added check for fontspec	62
\gls@protected@pagefmts: new	201	\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty	260
\gls@romanpage: new	202	\setglossarypreamble: new	51
\glsdefmain: added check for doc package	19	3.08a – 2013-08-30 altlist: updated altlist style to use \glossentry and \subglossentry	276
\glsorg@endtheglossary: new	6	inline: updated inline style to use \glossentry and \subglossentry	271
\glsorg@theglossary: new	6	list: updated list style to use \glossentry and \subglossentry	275
\PrintChanges: new	6	listdotted: updated listdotted style to use \glossentry and \subglossentry	277
3.05 – 2013-04-21 \@@do@esc@wrglossary: add Roman case. Fixed bugs in the else statements	205	3.08a – 2013-09-28 General: added nogroupskip key to \printglossary	226
\@gls@link: added check for “nohypertypes”	122	removed definition of \@glossaryentryfield	358
\gls@protected@pagefmts: added Roman to list	201	removed definition of \@glossarysubentryfield	358
\gls@Romanpage: new	202	\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort	102
\glsgetgrouplabel: fixed bug (typo in \equal)	233	updated for \glossentry	102
\glsmcols: new	295	\@glossaryentryfield: switched to \glossentry	101
mcolalttree: replaced ‘2’ with \glsmcols	299	\@glossarysubentryfield: switched to \subglossentry	101
mcolindex: replaced ‘2’ with \glsmcols	295	\@gls@sanitizedesc: deprecated	24
mcolindexspannav: replaced ‘2’ with \glsmcols	296	altlongragged4col: updated to use \glossentry and \subglossentry	293
mcoltree: replaced ‘2’ with \glsmcols	296		
mcoltreenoname: replaced ‘2’ with \glsmcols	298		
mcoltreespannav: replaced ‘2’ with \glsmcols	297		
\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		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	321	\subglossentry	308
\compatibleglossentry: new ..	230	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	231	\subglossentry	317
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	231	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	234	new	25
\glossentry: new	230	\@gls@default@value: new ..	76
\Glossentrydesc: new	231	\Glsentrydesc: made robust ..	167
\glossentrydesc: new	230	\Glsentrydescplural: made	
\Glossentryname: new	230	robust	167
\glossentryname: new	230	\Glsentryfirst: made robust ..	168
\Glossentrysymbol: new	231	\Glsentryfirstplural: made	
\glossentrysymbol: new	231	robust	169
\gls@assign@desc@field: new ..	24	\Glsentryfull: made robust ..	173
\gls@assign@descplural@field:		\Glsentryfullpl: made robust ..	173
new	25	\Glsentrylong: made robust ..	172
\gls@assign@field: new	83	\Glsentrylongpl: made robust ..	172
\gls@ifnotmeasuring: new ..	103	\Glsentryname: made robust ..	165
\glsaddallunused: new	176	\Glsentryplural: made robust ..	167
\glsexpandfields: new	83	\Glsentryshort: made robust ..	172
\glsnoexpandfields: new	83	\Glsentryshortpl: made robust ..	172
\glssee: made robust	209	\Glsentrysymbol: made robust ..	168
\glsseeformat: made robust ..	209	\Glsentrysymbolplural: made	
\glsseeitem: made robust	210	robust	168
\glsseelist: made robust	209	\Glsentrytext: made robust ..	167
\ifglsdessuppressed: new ..	68	\Glsentryuseri: made robust ..	170
\ifglshasdesc: new	67	\Glsentryuserii: made robust ..	170
\ifglshassymbol: new	68	\Glsentryuseriii: made robust ..	170
index: added paragraph break at		\Glsentryuseriv: made robust ..	171
end of environment	315	\Glsentryuserv: made robust ..	171
updated to use \glossentry		\Glsentryuservi: made robust ..	171
and \subglossentry	316	\glstextup: new	240
long: updated to use		\ifglshassymbol: changed test	
\glossentry and		to check for	
\subglossentry	278	\@gls@default@symbol	68
longragged: updated to use		3.10a – 2013-09-28	
\glossentry and		\gls@assign@type@field: new ..	24
\subglossentry	290	3.10a – 2013-10-13	
longragged3col: updated to use		\@gls@keymap: new	86
\glossentry and		\@gls@provide@newglossary:	
\subglossentry	291	new	71
\setglossarystyle: new	234	\@gls@writedef: new	86
\setglossentrycompatibility:		\@glsdefaultplural: Obsolete ..	80
new	231	\@glsnodesc: new	80
\subglossentry: new	232	\@print@glossary: Added	
		providecommand code to aux	
		file	214, 215

<code>\gls@defglossaryentry:</code>		
Changed to using		
<code>\@gls@default@value</code>	95	
new	95	
<code>\glswritedefhook:</code> new	94	
<code>\makeglossaries:</code> Added		
providecommand code to aux		
file	192	
<code>\new@glossaryentry:</code> new	84	
<code>\ns@newglossary:</code> added		
<code>\@gls@provide@newglossary</code>	73	
3.11a – 2013-10-15		
General: added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	156–163	
changed to just use		
<code>\Glsentrydescplural</code>	147	
changed to just use		
<code>\glsentrydescplural</code>	147	
changed to just use		
<code>\Glsentrydesc</code>	146	
changed to just use		
<code>\glsentrydesc</code>	146	
changed to just use		
<code>\Glsentryfirstplural</code>	144	
changed to just use		
<code>\glsentryfirstplural</code>	144	
changed to just use		
<code>\Glsentryfirst</code>	142	
changed to just use		
<code>\glsentryfirst</code>	142	
changed to just use		
<code>\Glsentryname</code>	145	
changed to just use		
<code>\glsentryname</code>	145	
changed to just use		
<code>\Glsentryplural</code>	143	
changed to just use		
<code>\glsentryplural</code>	143	
changed to just use		
<code>\Glsentrysymbolplural</code>	149	
changed to just use		
<code>\glsentrysymbolplural</code>	149	
changed to just use		
<code>\Glsentrysymbol</code>	148	
changed to just use		
<code>\glsentrysymbol</code>	148	
Changed to just use		
<code>\Glsentrytext</code>	141	
changed to just use		
<code>\glsentrytext</code>	141	
changed to just use		
<code>\Glsentryuseriii</code>	152	
changed to just use		
<code>\glsentryuseriii</code>	152	
changed to just use		
<code>\Glsentryuserii</code>	151	
changed to just use		
<code>\glsentryuserii</code>	151	
changed to just use		
<code>\Glsentryuseriv</code>	153	
changed to just use		
<code>\glsentryuseriv</code>	153	
changed to just use		
<code>\Glsentryuseri</code>	150	
changed to just use		
<code>\glsentryuseri</code>	150	
changed to just use		
<code>\Glsentryuservi</code>	155	
changed to just use		
<code>\glsentryuservi</code>	155	
changed to just use		
<code>\Glsentryuserv</code>	154	
changed to just use		
<code>\glsentryuserv</code>	154	
Now requires <code>textcase</code>	5	
<code>\@ACRlong:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	353	
<code>\@ACRshort:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	352	
<code>\@Acrlong:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	353	
<code>\@Acrshort:</code> added <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glsinsert</code> and		
<code>\glscustomtext</code>	351	
<code>\@GLS@:</code> add <code>\glslabel</code> ,		
<code>\glsifplural</code> , <code>\glscapscase</code> ,		
<code>\glscustomtext</code> and		
<code>\glsinsert</code>	136	
change to using <code>\glsentryfmt</code>		
style commands	136	

removed \MakeUppercase (now moved to \glentryfmt) ..	136	\@glsp10: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	137
\@GLSpl: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	139	change to using \glentryfmt style commands	137
change to using \glentryfmt style commands	139	acronymlists: replaced \@addtoacronymlists with \DeclareAcronymList	22
removed \MakeUppercase as now dealt with in \glentryfmt	139	compatible-3.07: new	38
\@Gls@: add \glslifplural, \glscapscase, \glscustomtext and \glinsert	136	\defgldisplay: obsoleted	118
change to using \glentryfmt style commands	136	\defgldisplayfirst: obsoleted	118
removed \makefirstuc (now dealt with in \glentryfmt)	136	\defglentryfmt: new	71
\@GLspl@: add \glslifplural, \glscapscase, \glscustomtext and \glinsert	138	\forglentries: replaced \ifx with \ifdefempty	64
change to using \glentryfmt style commands	138	\gl@assign@desc: new	94
removed \makefirstuc (now dealt with in \glentryfmt)	138	\gl@defglossaryentry: Fixed default counter if none supplied	98
\@acrlong: added \glslabel, \glslifplural, \glscapscase, \glinsert and \glscustomtext	352	\gl@doentryfmt: new	71
\@acrshort: added \glslabel, \glslifplural, \glscapscase, \glinsert and \glscustomtext	351	\gldisplay: obsoleted	118
\@gls@: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	135	\gldisplayfirst: obsoleted ..	117
change to using \glentryfmt style commands	135	\glsgentryfmt: new	112
\@gls@noexpand@fields: Fixed bug expand replaced with noexpand	81	\glsgetgroupitle: Added check in case non-Latin alphabet in use	233
\@gldisp: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	139	\glsglossarymark: replaced \MakeUppercase with \mfirstucMakeUppercase ..	51
change to using \glentryfmt style commands	139	\glnavigation: switched to using \@gls@getgroupitle	270
		\ifglshasdesc: replaced \ifdefempty with \ifcempty	67
		\ifglshaslong: new	68
		\ifglshasshort: new	68
		\ifglshassymbol: replaced \ifdefempty with \ifcempty	68
		\ifglused: replaced \ifthenelse with \ifbool ..	65
		\longnewglossaryentry: new ..	94
		\ns@newglossary: replaced \gldisplay and \gldisplayfirst with \glentryfmt	73
		numbers: new	40
		\SetCustomDisplayStyle: updated to use \defglentryfmt	261

\SetDefaultAcronymDisplayStyle:	\gls@assign@desc@field:
changed to use	changed to use
\defglsentryfmt 258	\glsetnoexpandfield 24
\SetDescriptionAcronymDisplayStyle:	\gls@assign@descplural@field:
updated to use	changed to use
\defglsentryfmt 260	\glsetnoexpandfield 25
\SetDescriptionDUAAcronymDisplayStyle:	\gls@assign@name@field:
updated to use	changed to use
\defglsentryfmt 259	\glsetnoexpandfield 25
\SetDescriptionFootnoteAcronymDisplayStyle:	\gls@assign@type@field:
updated to use	changed to use
\defglsentryfmt 259	\glsetexpandfield 24
\SetDUADisplayStyle: updated	\gls@checkseeallowed: new . . 77
to use \defglsentryfmt . . 261	\glsaddallunused: set default to
\SetFootnoteAcronymDisplayStyle:	\@glo@types 176
updated to use	\Glsentryfull: changed to use
\defglsentryfmt 260	\acrfullformat 173
\SetSmallAcronymDisplayStyle:	\glsetentryfull: changed to use
updated to use	\acrfullformat 173
\defglsentryfmt 260	\Glsentryfullpl: changed to use
\setupglossaries: new 42	\acrfullformat 173
\showglolong: new 266	\glsetentryfullpl: changed to use
\showgloshort: new 265	\acrfullformat 173
symbols: new 40	\glsglossarymark: renamed
3.12a – 2013-10-16	\glossarymark to
\gls@defglossaryentry: added	\glsglossarymark to avoid
\glslabel 95	conflict with memoir 51
\glsaddkey: new 88	\glsprestandardsort: new . . . 15
3.13a – 2013-11-05	\glsetexpandfield: new 24
\@gls@assign@symbol@field:	\glsetnoexpandfield: new . . 24
changed to use	long: switched to
\glsetnoexpandfield 25	\tabularnewline 278, 279
\@gls@assign@symbolplural@field:	long3col: switched to
changed to use	\tabularnewline 280
\glsetnoexpandfield 25	long3colheader: switched to
\@gls@link: removed \relax . . 123	\tabularnewline 281
\@gls@notranslatorhook: new . . 29	long3colheaderborder: switched
\@gls@setupsort@standard:	to \tabularnewline 281
moved \@gls@santizesort to	long4col: switched to
\glsprestandardsort 16	\tabularnewline 281
altsuper4colheader: switched to	long4colheader: switched to
\tabularnewline 306	\tabularnewline 282
altsuper4colheaderborder:	longheader: switched to
switched to	\tabularnewline 279
\tabularnewline 307	longheaderborder: switched to
\glossarysection: changed	\tabularnewline 279
\glossarymark to	see: added
\glsglossarymark 51	\gls@checkseeallowed 77
	super: switched to
	\tabularnewline 301

super3col: switched to		\@acrfull: now using	
\tabularnewline	303	\acrfullfmt	242
super3colheader: switched to		\@gls@indexdef: new	41
\tabularnewline	304	\@gls@numbersdef: new	40
super4col: switched to		\@gls@symbolsdef: new	40
\tabularnewline	305	\ACRfullfmt: new	241
super4colheader: switched to		\Acrfullfmt: new	241
\tabularnewline	305	\acrfullfmt: new	241
super4colheaderborder:		\ACRfullplfmt: new	241
switched to		\Acrfullplfmt: new	241
\tabularnewline	306	\acrfullplfmt: new	241
superheader: switched to		\acronymentry: new	245
\tabularnewline	302	dua: new	252
superheaderborder: switched to		dua-desc: new	254
\tabularnewline	302	footnote: new	254
ucmark: added check for memoir	14	footnote-desc: new	256
3.14a – 2013-11-12		footnote-sc: new	255
General: new	325	footnote-sc-desc: new	256
\@glswritefiles: renamed		footnote-sm: new	255
\glswritefiles to		footnote-sm-desc: new	256
\@glswritefiles and used		\Genacrfullformat: new	117
“savewrites” option to set		\genacrfullformat: new	117
\glswritefiles	198	\GenericAcronymFields: new	245
acronyms: new	20	\Genplacrfullformat: new	117
compatible-2.07: added check		\genplacrfullformat: new	117
for 2.07 before setting 3.07		\Glsentryfull: bug fix: added	
compatibility	38	missing \acronymfont	173
\gls@defglossaryentry: added		\Glsentryfull: bug fix: added	
check for existence of default		missing \acronymfont	173
glossary	96	\Glsentryfullpl: bug fix: added	
set the default for firstplural to		missing \acronymfont	173
be the value of plural	98	\Glsentryfullpl: bug fix: added	
\longprovideglossaryentry:		missing \acronymfont	173
new	95	\glsgenacfmt: new	115
notranslate: new	30	\GlsUseAcrEntryDisplayStyle:	
\provideglossaryentry: new	83	new	247
xindygloss: new	34	\GlsUseAcrStyleDefs: new	247
4.0 – 2013-11-14		index: new	40
\gls@defglossaryentry: added		long-sc-short: new	249
check for first key	98	long-sc-short-desc: new	250
super: fixed typo in		long-short: new	247
\subglossentry		long-short-desc: new	250
(\glossentrydesc)	301	long-sm-short: new	249
4.01 – 2013-11-16		long-sm-short-desc: new	251
General: fixed non-value options		\newacronymstyle: new	246
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	160–163	sc-short-long: new	249

sc-short-long-desc: new	251	moved \@gls@link@opts and \@gls@link@label to \@gls@link	122
\setacronymstyle: new	246	\@gls@writedef: added \glsdetoklabel	86
\SetGenericNewAcronym: new . .	244	\@glsdisp: removed \glslabel (defined in \@gls@link) . .	139
short-long: new	248	\@glspl@: removed \glslabel (defined in \@gls@link) . .	137
short-long-desc: new	251	\@printglossary: added \glsdetoklabel	213
sm-short-long: new	249	\compatibleglossentry: added \glsdetoklabel	333
sm-short-long-desc: new	251	\compatiblesubglossentry: added \glsdetoklabel . . .	333
xindynoglsnumbers: new	34	dua: fixed bug in \acrfullfmt .	253
4.02 – 2013-13-05		fixed bug in \Acrfullplfmt .	253
\makeglossaries: made preamble only	194	fixed bug in \acrfullplfmt .	253
4.03 – 2014-01-17		redefined to use accessibility information	363
General: changed default to \@empty instead of \@relax . .	38	dua-desc: commented spurious EOL	254
4.03 – 2014-01-20		redefined to use accessibility information	365
General: removed \glslabel (defined in \@gls@link) . .	156	footnote: redefined to use accessibility information . .	366
\@do@esc@wrglossary: added \glsdetoklabel	206	footnote-desc: redefined to use accessibility information . .	368
\@do@noesc@wrglossary: added \glsdetoklabel	204	footnote-sc: redefined to use accessibility information . .	367
\@ACRlong: removed \glslabel (defined in \@gls@link) . .	353	footnote-sc-desc: redefined to use accessibility information	368
\@ACRshort: removed \glslabel (defined in \@gls@link) . .	352	footnote-sm: redefined to use accessibility information . .	368
\@Acrlong: removed \glslabel (defined in \@gls@link) . .	353	footnote-sm-desc: redefined to use accessibility information	368
\@Acrshort: removed \glslabel (defined in \@gls@link) . .	351	\Genacrfullformat: redefined to use accessibility information	351
\@GLS@: removed \glslabel (defined in \@gls@link) . .	136	\genacrfullformat: redefined to use accessibility information	350
\@GLSpl: removed \glslabel (defined in \@gls@link) . .	139	\Genplacrfullformat: redefined to use accessibility information	351
\@Gls@: removed \glslabel (defined in \@gls@link) . .	136	\genplacrfullformat: redefined to use accessibility information	351
\@Gls@entry@field: new	164	\glossentryname: added \glsdetoklabel	230
\@Glspl@: removed \glslabel (defined in \@gls@link) . .	138	\gls@defglossaryentry: added \glsdetoklabel	95
\@acrlong: removed \glslabel (defined in \@gls@link) . .	352		
\@acrshort: removed \glslabel (defined in \@gls@link) . .	351		
\@gls@: removed \glslabel (defined in \@gls@link) . .	135		
\@gls@access@display: new . .	342		
\@gls@entry@field: new	163		
\@gls@fetchfield: new	87		
\@gls@field@link: new	140		
\@gls@link: added \glsdetoklabel	122		

replaced #1 with \@glo@label	96	\glslocalunset: added	
replaced \ifthenelse with		\glsdetoklabel	104
\ifdefequal	97	\glsmoveentry: added	
\glsadd: added \glsdetoklabel	175	\glsdetoklabel	101
\glsaddkey: switched to using		replaced \ifthenelse with	
\@gls@field@link	89	\ifdefequal	101
\glsdetoklabel: new	65	\glsrefentry: added	
\glsdisplaynumberlist: added		\glsdetoklabel	228
\glsdetoklabel	174	\glsreset: added	
\glsdoifexistsorwarn: new	66	\glsdetoklabel	104
\glsentryaccess: switched to		\glsseelist: added	
using \@gls@entry@field	338	\expandafter commands	209
\glsentrydescaccess: switched		\glsstepentry: added	
to using \@gls@entry@field	339	\glsdetoklabel	228
\glsentrydescpluralaccess:		\glsstepsubentry: added	
switched to using		\glsdetoklabel	228
\@gls@entry@field	339	\glsunset: added	
\glsentryfirstaccess: switched		\glsdetoklabel	104
to using \@gls@entry@field	339	\ifglsdescsuppressed: added	
\glsentryfirstplural: added		\glsdetoklabel	68
\glsdetoklabel	169	fixed typo	68
\glsentrylongaccess: switched		\ifglsentryexists: added	
to using \@gls@entry@field	340	\glsdetoklabel	65
\glsentrylongpluralaccess:		\ifglschaschildren: added	
switched to using		\glsdetoklabel	67
\@gls@entry@field	340	\ifglschasdesc: added	
\glsentrypluralaccess:		\glsdetoklabel	67
switched to using		\ifglschasfield: new	69
\@gls@entry@field	339	\ifglschaslong: added	
\glsentryshortaccess: switched		\glsdetoklabel	68
to using \@gls@entry@field	339	\ifglschasparent: added	
\glsentryshortpluralaccess:		\glsdetoklabel	67
switched to using		\ifglschasshort: added	
\@gls@entry@field	339	\glsdetoklabel	68
\glsentrysymbolaccess:		\ifglschassymbol: added	
switched to using		\glsdetoklabel	68
\@gls@entry@field	339	replaced \ifcseempty with	
\glsentrysymbolpluralaccess:		\ifdefempty and replaced	
switched to using		\ifx with \ifdefequal	68
\@gls@entry@field	339	\ifglsused: added	
\glsentrytextaccess: switched		\glsdetoklabel	65
to using \@gls@entry@field	338	long-sc-short-desc: redefined	
\glsacacfmt: redefined to use		to use accessibility	
accessibility information	348	information	361
\glsacentryfmt: redefined to		long-short: redefined to use	
use accessibility information	346	accessibility information	360
\gshyperlink: added		long-short-desc: redefined to	
\glsdetoklabel	174	use accessibility information	361
\glslocalreset: added			
\glsdetoklabel	104		

long-sm-short-desc: redefined to use accessibility information	362	\showgloplural: added \glsdetoklabel	263
\renewacronymstyle: new	246	\showglopluralaccess: added \glsdetoklabel	370
sc-short-long-desc: redefined to use accessibility information	362	\showgloshort: added \glsdetoklabel	265
short-long: commented spurious EOL	249	\showgloshortaccess: added \glsdetoklabel	371
redefined to use accessibility information	360	\showgloshortpluralaccess: added \glsdetoklabel . . .	371
short-long-desc: redefined to use accessibility information	362	\showglosort: added \glsdetoklabel	265
\showglocounter: added \glsdetoklabel	263	\showglosymbol: added \glsdetoklabel	265
\showglodesc: added \glsdetoklabel	265	\showglosymbolaccess: added \glsdetoklabel	370
\showglodescaccess: added \glsdetoklabel	370	\showglosymbolplural: added \glsdetoklabel	265
\showglodescplural: added \glsdetoklabel	265	\showglosymbolpluralaccess: added \glsdetoklabel . . .	370
\showglodescpluralaccess: added \glsdetoklabel . . .	370	\showglotext: added \glsdetoklabel	262
\showglofirst: added \glsdetoklabel	263	\showglotextaccess: added \glsdetoklabel	370
\showglofirstaccess: added \glsdetoklabel	370	\showglotype: added \glsdetoklabel	263
\showglofirsttpl: added \glsdetoklabel	263	\showglouseri: added \glsdetoklabel	263
\showglofirstpluralaccess: added \glsdetoklabel . . .	370	\showglouserii: added \glsdetoklabel	264
\showgloflag: added \glsdetoklabel	266	\showglouseriii: added \glsdetoklabel	264
\showgloindex: added \glsdetoklabel	266	\showglouseriv: added \glsdetoklabel	264
\showglolevel: added \glsdetoklabel	262	\showglouserv: added \glsdetoklabel	264
\showglolong: added \glsdetoklabel	266	\showglouservi: added \glsdetoklabel	264
\showglolongaccess: added \glsdetoklabel	371	sm-short-long-desc: redefined to use accessibility information	363
\showglolongpluralaccess: added \glsdetoklabel . . .	371	4.04 – 2014-03-04 \@gls@getcounterprefix: added warning if no prefix can be formed	208
\showgloname: added \glsdetoklabel	264	4.04 – 2014-03-06 General: added sort key to printgloss group	226
\showglonameaccess: added \glsdetoklabel	370	\@@gls@noidx@nosanitizesort: new	26
\showgloparent: added \glsdetoklabel	262		

<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> ...	224
<code>\@glo@addchildren: new</code>	216	<code>\glsnoidxdisplayloclisthandler:</code>	
<code>\@glo@do@sortentries: new</code> ..	216	new	224
<code>\@glo@grabfirst: new</code>	221	<code>\glsnoidxloclist: new</code>	223
<code>\@glo@sortedinsert: new</code>	217	<code>\glsnoidxloclisthandler: new</code>	223
<code>\@glo@sortentries: new</code>	215	<code>\glsnoidxstripaccents: new</code> ..	26
<code>\@glo@sorthandler@case: new</code> .	218	<code>\makeglossaries: Moved</code>	
<code>\@glo@sorthandler@letter:</code>		definition of <code>\glswrite</code> to	
new	217	<code>\makeglossaries</code>	192
<code>\@glo@sorthandler@nocase:</code>		<code>\makenoidxglossaries: new</code> ..	194
new	218	<code>\printglossary: changed to use</code>	
<code>\@glo@sorthandler@word: new</code> .	217	new <code>\@printglossary</code>	211
<code>\@glo@sortmacro@case: new</code> ..	219	<code>\printnoidxglossaries: new</code> ..	212
<code>\@glo@sortmacro@def: new</code> ...	220	<code>\printnoidxglossary: new</code> ...	212
<code>\@glo@sortmacro@def@do: new</code> .	220	<code>\showgloloclist: new</code>	266
<code>\@glo@sortmacro@letter: new</code> .	218	<code>\warn@noprintglossary:</code>	
<code>\@glo@sortmacro@nocase: new</code> .	219	Activate warning in	
<code>\@glo@sortmacro@standard:</code>		<code>\makeglossaries</code>	210
new	219	<code>\writeist: checked for definition</code>	
<code>\@glo@sortmacro@use: new</code> ...	220	of <code>\glswrite</code>	179, 183
<code>\@glo@sortmacro@word: new</code> ..	218	4.06 – 2014-03-12	
<code>\@gls@noidx@do: new</code>	222	General: added	
<code>\@gls@noidx@getgrouptitle:</code>		<code>\glsifhyper</code>	156–163
new	233	<code>\@GLS@: added \glsifhyper</code> ...	136
<code>\@gls@noref@warn: new</code>	198	<code>\@GLSpl: added \glsifhyper</code> ..	139
<code>\@gls@reference: new</code>	224	<code>\@Gls@: added \glsifhyper</code> ...	136
<code>\@gls@warnonglossdefined: new</code>	24	<code>\@Glspl@: added \glsifhyper</code> ..	138
<code>\@gls@warnontheglossdefined:</code>		<code>\@gls@: added \glsifhyper</code> ...	135
new	24	<code>\@gls@numbersdef: added hook</code>	
<code>\@no@makeglossaries: new</code> ...	198	to set toc title	40
<code>\@print@glossary: new</code>	214	<code>\@gls@symbolsdef: added hook</code>	
<code>\@print@noidx@glossary: new</code> .	220	to set toc title	40
<code>\@printgloss@setsort: new</code> ..	212	<code>\@glsdisp: added \glsifhyper</code>	139
<code>\@printglossary: new</code>	212	<code>\@glspl@: added \glsifhyper</code> ..	137
<code>alttree: moved hangindent and</code>		acronym: added hook to set toc	
<code>parindent</code> assignments outside		title	20
level test	321	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	333	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	333	unstarred version	52
<code>\glo@grabfirst: new</code>	221	<code>\@gls@noidx@do: added \global</code>	
		in case it's used in a	
		tabular-like style	222

<code>\Acrfullplfmt</code> : fixed no case		removed <code>\@GLSsymbol</code>	148
change bug	241	removed <code>\@GLSSymbol</code>	148
<code>\glsletentryfield</code> : new	163	removed <code>\@sglssymbol</code>	147
4.08 – 2014-07-30		removed <code>\@GLStext</code>	141
General: added entrycounter		removed <code>\@sGlstext</code>	141
option to <code>printgloss</code> family	226	removed <code>\@sglstext</code>	140
added nopostdot option to		removed <code>\@GLSuseriii</code>	152
<code>printgloss</code> family	226	removed <code>\@Glsuseriii</code>	152
added subentrycounter option		removed <code>\@sglsuseriii</code>	151
to <code>printgloss</code> family	226	removed <code>\@GLSuserii</code>	151
explicitly initialise hyper key	119	removed <code>\@Glsuserii</code>	151
moved <code>\glsifhyper</code>	156–163	removed <code>\@sglsuserii</code>	150
removed <code>\@sACRlongpl</code>	163	removed <code>\@GLSuseriv</code>	153
removed <code>\@sAcrlongpl</code>	162	removed <code>\@Glsuseriv</code>	153
removed <code>\@sacrlongpl</code>	161	removed <code>\@sglsuseriv</code>	152
removed <code>\@sACRlong</code>	161	removed <code>\@GLSuseri</code>	150
removed <code>\@sAcrlong</code>	160	removed <code>\@sGlsuseri</code>	150
removed <code>\@sacrlong</code>	159	removed <code>\@sglsuseri</code>	149
removed <code>\@sACRshortpl</code>	159	removed <code>\@GLSuservi</code>	155
removed <code>\@sAcrshortpl</code>	158	removed <code>\@Glsuservi</code>	155
removed <code>\@sacrshortpl</code>	157	removed <code>\@sglsuservi</code>	154
removed <code>\@sACRshort</code>	157	removed <code>\@GLSuserv</code>	154
removed <code>\@sAcrshort</code>	156	removed <code>\@sGlsuserv</code>	154
removed <code>\@sacrshort</code>	155	removed <code>\@sglsuserv</code>	153
removed <code>\@sgls@link</code>	120	removed <code>\@GLS</code>	136
removed <code>\@sGLSdescplural</code>	147	removed <code>\@Gls</code>	135
removed <code>\@sGLSdescplural</code>	147	removed <code>\@sgls</code>	134
removed <code>\@sglsdescplural</code>	146	removed <code>\@thirdofthree</code>	
removed <code>\@sGLSdesc</code>	146	(defined in kernel)	134
removed <code>\@sGlsdesc</code>	146	removed <code>sPGLS</code>	331
removed <code>\@sglsdesc</code>	145	removed <code>sPgls</code>	329
removed <code>\@sglsdisp</code>	139	removed <code>spgls</code>	328
removed <code>\@sGLSfirstplural</code>	144	removed <code>sPGLSpl</code>	331
removed <code>\@sGlsfirstplural</code>	144	removed <code>sPglspl</code>	330
removed <code>\@sglsfirstplural</code>	143	removed <code>spglspl</code>	329
removed <code>\@sGLSfirst</code>	142	<code>\@ACRlong</code> : added	
removed <code>\@sGlsfirst</code>	142	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sglsfirst</code>	141	353
removed <code>\@sGLSname</code>	145	<code>\@ACRshort</code> : added	
removed <code>\@sGlsname</code>	145	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sglsname</code>	144	352
removed <code>\@sGLSplural</code>	143	<code>\@Acrlong</code> : added	
removed <code>\@sGlsplural</code>	143	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sglsplural</code>	142	352
removed <code>\@sGLSpl</code>	138	<code>\@Acrshort</code> : added	
removed <code>\@sGlspl</code>	138	<code>\do@gls@link@checkfirsthyper</code>	
removed <code>\@sglspl</code>	137	351
removed <code>\@sGLSsymbolplural</code>	149	<code>\@GLS@</code> : moved <code>\glsifhyper</code>	136
removed <code>\@sGlsymbolplural</code>	149	moved check for first use to	
removed <code>\@sglsymbolplural</code>	148	<code>\@gls@link</code>	136

\@GLSspl: moved \glsifhyper ..	139	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	243
\@gls@link	139	switched to using	
\@Gls@: moved \glsifhyper ...	136	\@gls@hyp@opt	243
moved check for first use to		\Acrfullpl: removed	
\@gls@link	136	\s@Acrfullpl	243
\@Glspl@: moved \glsifhyper .	138	switched to using	
moved check for first use to		\@gls@hyp@opt	243
\@gls@link	138	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	243
\do@gls@link@checkfirsthyper		switched to using	
.....	352	\@gls@hyp@opt	242
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	160
.....	351	\Acrlong: switched to using	
\@closegls: new	189	\@gls@hyp@opt	160
\@gls@: moved \glsifhyper ...	135	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	159
\@gls@link	135	\ACRlongpl: switched to using	
\@gls@automake: new	189	\@gls@hyp@opt	162
\@gls@doautomake: new	37	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	162
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	161
.....	140	\ACRshort: switched to using	
\@gls@forbidtextext: new	71	\@gls@hyp@opt	157
\@gls@hyp@opt: new	120	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	156
redundancy	122	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	155
\glstype	122	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	158
new	121	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	139	\@gls@hyp@opt	158
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	139	\@gls@hyp@opt	157
\@glspl@: moved \glsifhyper .	137	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	137	instead of symbol	293
\@ignored@glossaries: new ..	75	automake: new	36
\ACRfull: removed \s@ACRfull	242	\forallacronyms: new	63
switched to using		\GLS: switched to using	
\@gls@hyp@opt	242	\@gls@hyp@opt	136
\Acrfull: removed \s@Acrfull	242	\Gls: switched to using	
switched to using		\@gls@hyp@opt	135
\@gls@hyp@opt	242	\gls: switched to using	
\acrfull: removed \s@acrfull	242	\@gls@hyp@opt	134
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	241	check for ignored glossary ..	96
		\gls@istfilebase: new	48

\glsaddkey: removed		
\@sGLS@user@<key>	90	
removed \@sGLS@user@<key>	90	
removed \@sgls@user@<key>	89	
switched to using		
\@gls@hyp@opt	89, 90	
\GLSdesc: switched to using		
\@gls@hyp@opt	146	
\Glsdesc: switched to using		
\@gls@hyp@opt	146	
\glsdesc: switched to using		
\@gls@hyp@opt	145	
\GLSdescplural: switched to		
using \@gls@hyp@opt	147	
\Glsdescplural: switched to		
using \@gls@hyp@opt	147	
\glsdescplural: switched to		
using \@gls@hyp@opt	146	
\glsdisablehyper: added		
\KV@glslink@hyperfalse to		
definition	134	
\glsdisp: switched to using		
\@gls@hyp@opt	139	
\glsdohyperlink: new	133	
\glsdohypertarget: new	133	
\glsenablehyper: added		
\KV@glslink@hypertrue to		
definition	134	
\GLSfirst: switched to using		
\@gls@hyp@opt	142	
\Glsfirst: switched to using		
\@gls@hyp@opt	142	
\glsfirst: switched to using		
\@gls@hyp@opt	141	
\GLSfirstplural: switched to		
using \@gls@hyp@opt	144	
\Glsfirstplural: switched to		
using \@gls@hyp@opt	144	
\glsfirstplural: switched to		
using \@gls@hyp@opt	143	
\glsifhyper: deprecated	119	
\glslink: switched to using		
\@gls@hyp@opt	120	
\glslinkcheckfirsthyperhook:		
new	121	
\glslinkvar: new	119	
\GLSname: switched to using		
\@gls@hyp@opt	145	
\Glsname: switched to using		
\@gls@hyp@opt	145	
\glsname: switched to using		
\@gls@hyp@opt	144	
\GLSpl: switched to using		
\@gls@hyp@opt	138	
\Glspl: switched to using		
\@gls@hyp@opt	137	
\glspl: switched to using		
\@gls@hyp@opt	137	
\GLSplural: switched to using		
\@gls@hyp@opt	143	
\Glsplural: switched to using		
\@gls@hyp@opt	143	
\glsplural: switched to using		
\@gls@hyp@opt	142	
\glsspace: new	241	
\GLSsymbol: switched to using		
\@gls@hyp@opt	148	
\Glsymbol: switched to using		
\@gls@hyp@opt	148	
\glssymbol: switched to using		
\@gls@hyp@opt	147	
\GLSsymbolplural: switched to		
using \@gls@hyp@opt	149	
\Glsymbolplural: switched to		
using \@gls@hyp@opt	149	
\glssymbolplural: switched to		
using \@gls@hyp@opt	148	
\GLStext: switched to using		
\@gls@hyp@opt	141	
\Glstext: switched to using		
\@gls@hyp@opt	141	
\glstext: switched to using		
\@gls@hyp@opt	140	
\glstreenamefmt: new	314	
\GLSuseri: switched to using		
\@gls@hyp@opt	150	
\Glsuseri: switched to using		
\@gls@hyp@opt	150	
\glsuseri: switched to using		
\@gls@hyp@opt	149	
\GLSuserii: switched to using		
\@gls@hyp@opt	151	
\Glsuserii: switched to using		
\@gls@hyp@opt	151	
\glssuserii: switched to using		
\@gls@hyp@opt	150	
\GLSuseriii: switched to using		
\@gls@hyp@opt	152	
\Glsuseriii: switched to using		
\@gls@hyp@opt	152	

\glsuseriii: switched to using		\@gls@glossary: Renamed	
\@gls@hyp@opt	151	\@glossary to	
\GLSuseriv: switched to using		\@gls@glossary	200
\@gls@hyp@opt	153	alttree: moved space after	
\GLSuseriv: switched to using		symbol	321, 322
\@gls@hyp@opt	153	\glspercentchar: new	177
\glsuseriv: switched to using		\glstildechar: new	177
\@gls@hyp@opt	152	4.11 – 2014-09-01	
\GLSuseriv: switched to using		\@do@esc@wrglossary: added	
\@gls@hyp@opt	154	hook	206
\GLSuseriv: switched to using		\gls@wrglossary: renamed from	
\@gls@hyp@opt	154	\@wrglossary to	
\glsuseriv: switched to using		\gls@wrglossary	200
\@gls@hyp@opt	153	\glsaddprotectedpagefmt: new	202
\GLSuseriv: switched to using		\glsbackslash: new	177
\@gls@hyp@opt	155	sanitize: none option	29
\GLSuseriv: switched to using		4.12 – 2014-11-22	
\@gls@hyp@opt	155	\@gls@addpredefinedattributes:	
\glsuseriv: switched to using		Added glsignore attribute	57
\@gls@hyp@opt	154	\@gls@adjustmode: new	176
\ifignoredglossary: new	75	\@gls@notranslatorhook:	
\newglossary: added starred		removed	29
version	72	\@gls@toc: added \protect to	
\newignoredglossary: new	74	\numberline	54
\ns@newglossary: added		\@gls@usetranslator: new	29
\@gls@type@{name}@log	72	\glsacrpluralsuffix: new	45
new	72	\glsadd: added check for vertical	
\p@gls@hyp@opt: new	120	mode	175
\PGLS: changed to use		\glsaddallunused: replaced	
\@gls@hyp@opt	331	@gobble with glsignore	176
\Pgls: changed to use		\glsifusedtranslatordict: new	29
\@gls@hyp@opt	329	\glsignore: new	176
\pgls: changed to use		\glsupacrpluralsuffix: new	45
\@gls@hyp@opt	328	\ProvidesGlossariesLang: new	45
\PGLSpl: changed to use		\RequireGlossariesLang: new	45
\@gls@hyp@opt	331	4.13 – 2015-02-03	
\Pglspl: changed to use		\indexspace: new	274, 295, 314
\@gls@hyp@opt	330	4.14 – 2015-02-28	
\pglspl: changed to use		\@gls@localreset: new	105
\@gls@hyp@opt	329	\@gls@localunset: new	105
\s@gls@hyp@opt: new	120	\@gls@reset: new	105
\s@newglossary: new	72	\@gls@unset: new	105
4.09 – 2014-08-12		\@newglossaryentry@defcounters:	
\glsaddkey: fixed bug in user		new	106
commands	89	\@cGls: new	110
4.10 – 2014-08-27		\@cGls@: new	110
\@Gls@acrentryname: new	165	\@cGlspl@: new	111
\@Gls@entryname: new	165	\@cgls: new	109
		\@cgls@: new	110
		\@cglspl: new	111

\@cglspl@: new	111	\@GLSpl: added	
\@gls@entry@count: new	109	\glspostlinkhook	139
\@gls@increment@currcount:		\@Gls@: added	
new	109	\glspostlinkhook	136
\@gls@local@increment@currcount:		\@GLspl@: added	
new	109	\glspostlinkhook	138
\@gls@write@entrycounts: new	109	\@acrlong: added	
\@glslocalreset: new	105	\glspostlinkhook	352
\@glslocalunset: new	105	\@acrshort: added	
\@glsreset: new	105	\glspostlinkhook	351
\@glsunset: new	105	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	135
new	101	\@gls@@link: added	
\cGls: new	110	\glspostlinkhook	121
\cgls: new	109	\@gls@field@link: added	
\cGlsformat: new	110	\glspostlinkhook	140
\cglsformat: new	110	\@gls@link: moved definition of	
\cGlspl: new	111	\glsifhyperon outside of this	
\cglspl: new	110	macro	123
\cGlsplformat: new	111	\@glsdisp: added	
\cglsplformat: new	111	\glspostlinkhook	140
\gls@defdocnewglossaryentry:		\@GLspl@: added	
new	83	\glspostlinkhook	137
\glsenableentrycount: new	107	\glsacspace: new	248
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	104	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	175
\@glslocalunset	104	\glsfielddef: new	92
\glsreset: switched to		\glsfieldedef: new	91
\@glsreset	104	\glsfieldfetch: new	92
\glsunset: switched to		\glsfieldgdef: new	91
\@glsunset	104	\glsfieldxdef: new	91
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	122
\@glo@type with \glstype	162	\glslinkpostsetkeys: new	122
4.16 – 2015-06-18		\glspostlinkhook: new	121
\glsaddstoragekey: new	87	\glswriteentry: new	201
4.16 – 2015-07-08		\ifglsfieldcseq: new	94
General: added		\ifglsfielddefeq: new	93
\glspostlinkhook	156–163	\ifglsfieldefeq: new	93
\@ACRlong: added		long-sp-short: new	247
\glspostlinkhook	353	long-sp-short-desc: new	250
\@ACRshort: added		\showglofield: new	266
\glspostlinkhook	352	4.18 – 2015-09-09	
\@Acrlong: added		General: split mfirstuc into	
\glspostlinkhook	353	separate bundle	4
\@Acrshort: added		4.19 – 2015-10-31	
\glspostlinkhook	352	\glstreenamebox: new	320
\@GLS@: added			
\glspostlinkhook	137		

4.19 – 2015-11-22		
\@gls@link@nocheckfirsthyper:		\glsentrytitlecase: new 169
new 140		\glsfindwidesttoplevelname:
\@gls@preglossaryhook: new . 212		new 320
\@printglossary: added		\glslistgroupheaderfmt: new . 274
\@gls@preglossaryhook . . 214		\glslistnavigationitem: new . 274
\do@glsglisablehyperinlist:		\glstreegroupheaderfmt: new . 314
new 122		\glstreenavigationfmt: new . 314
\doifglossarynoexistsordo:		\ifglswrallowprimitivemods:
new 66		new 204
\gls@gobbleopt: new 72		list: fixed missing space before
\glsglsoifexistsordo: new 66		description 275
4.20 – 2015-11-30		long: fixed typo in
\@gls@link: added		\glossentrydesc 278
\@gls@setdefault@glslink@opts		mcolaltnreespannav: new 300
. 122		mcolindexspannav: new 296
added \glsglsonohyperlink		mcoltreenonamespannav: new . 298
when hyperlink is suppressed 123		mcoltreespannav: new 297
\@gls@setdefault@glslink@opts:		super4col: fixed bug in
new 122		\glossentry 305
\gls@checkseeallowed@preambleonly:	4.23 – 2016-04-30	
new 78		altlongragged4col: check for
\glsglsonohyperlink: new 133		nogroupskip changed 293
4.21 – 2016-01-24		altsuperragged4col: check for
General: changed checkfirsthyper		nogroupskip changed 312
assignment 156–163		\glscurrentfieldvalue: new . 70
\@printglossary: warn if no		\ifglshasfield: added
style has been set 212		\glscurrentfieldvalue 69, 70
altlong4col-booktabs: new . . 286		long: check for nogroupskip
altlongragged4col-booktabs:		changed 279
new 287		long-booktabs: check for
\glsglLtpenaltycheck: new 288		nogroupskip changed 285
\glsglpatchLtoutput: new 288		long3col: check for nogroupskip
\glsglpenaltygroupskip: new . . 288		changed 280
long-booktabs: new 284		long3col-booktabs: check for
long3col-booktabs: new 285		nogroupskip changed 285
long4col-booktabs: new 285		long4col: check for nogroupskip
longragged-booktabs: new . . . 286		changed 282
longragged3col-booktabs: new 287		long4col-booktabs: check for
\setglossarystyle: set default		nogroupskip changed 286
style if not already set 234		longragged: check for
4.22 – 2016-04-19		nogroupskip changed 290
\@@do@esc@wrglossary: added		longragged3col: check for
check for \@arabic 205		nogroupskip changed 291
added test to allow temporary		super: check for nogroupskip
primitive modifications and		changed 301
added arabic case 205		super3col: check for nogroupskip
\gls@arabicpage: new 202		changed 303
\gls@protected@pagefmts:		super4col: check for nogroupskip
added arabic to list 201		changed 305

superragged: check for nogroupskip changed	309	outside of theglossary environment	221
superragged3col: check for nogroupskip changed	311	\glsnavhyperlinkname: new	269
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	187	\@glo@autosee: new	100
\@gls@glossary: added check for debug mode	200	\@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	223
\glsnomakeindexwarning: new	54	\@gls@setupsort@def: added check for register	17
\GlsSetQuote: new	184	\@gls@setupsort@none: new	18
\GlsSetWriteIstHook: new	184	\@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 \ifundefined to \ifcsundef	345	\gls@xr@key: new	77
\@gls@enablesavenonumberlist: new	79	\GlsAddXdyLocation: bug fix: changed #1 to #2	60
\@gls@initnonumberlist: new	78, 79	\glsnoidxstripaccents: added \a	26
\@gls@savenonumberlist: new	78	added \TH, \dh and \DH	27
4.26 – 2016-10-12		4.31 – 2017-08-10	
\@glossary@default@style: added check for classicthesis	11	nolist: added check for “list” style	13
\glstreechildpredesc: new	315	4.31 – 2017-09-10	
\glstreeitem: new	315	style: changed \renewcommand to \def	11
\glstreepredesc: new	315	4.32 – 2017-08-24	
\glstreesubitem: new	315	\@glsnavhypertarget: new	269
\glstreesubsubitem: new	315	\@glsshowtarget: new	8
mcolindex: replaced \@idxitem with \glstreeitem	295	\glsshowtarget: new	7
mcolindexspannav: replaced \@idxitem with \glstreeitem	296	4.33 – 2017-09-20	
4.28 – 2017-01-07		\@do@esc@wrglossary: added \gls@the and \gls@number	205
\glspatchtabularx: new	103	renamed from \@do@esc@wrglossary	204
4.29 – 2017-01-19		\@do@noesc@wrglossary: new	203
General: added check for \@glxtr@doaccsupp	333	\@do@wrglossary: changed to check for esclocations	203
\@gls@noidx@do: current letter group assignment made global	223	\@gls@missinglang@warn: new	23
\@print@noidx@glossary: moved definition of \@gls@currentlettergroup		esclocations: new	12
		\GlsSetXdyFirstLetterAfterDigits: added starred version	177
		\GlsSetXdyNumberGroupOrder: new	178

4.34 – 2017-11-03		\glssubentryitem: bug fix: move conditional inside command	229
\glsshowtarget: modified to check for math mode and inner	7	\showglonameaccess: bug fix: corrected field (was showing text access field)	370
mcolalttreespannav: removed spurious space	300	4.40 – 2018-06-01	
4.35 – 2017-11-14		\istfile: changed \def to \providecommand	198
\glsadd: added \@gls@setsort (in case of sort=use)	175	\makenoidxglossaries: false	195
4.36 – 2018-03-07		4.41 – 2018-07-23	
\@gls@glossary: removed \index	200	General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr	225
4.37 – 2018-04-07		\@gls@override@glossary: new	39
\gls@begindocdefs: new	85	debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr	7
4.38 – 2018-05-10		\glossary: added warning	39
\@gls@define@glossaryentrycounter: added check for existence of glossaryentry counter	14	\gls@original@glossary: new	38
new	14	\gls@original@makeglossary: new	38
\@gls@define@glossarysubentrycounter: new	15	kernelglossredefs: new	39
prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space	15	\makeglossaries: removed redefinition of \makeglossary	193
\glsaccsupp: added braces around actual text argument	342	\makeglossary: added warning	38
\glsentrycounterlabel: bug fix: move conditional inside command	228	nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr	78
\GlsEntryCounterLabelPrefix: new	226	numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr	10
\glsentryitem: bug fix: move conditional inside command	229	seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr	9
\glsrefentry: bug fix: move conditional inside command	228	translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr	30
\glsresetentrycounter: bug fix: move conditional inside command	228	4.42 – 2019-01-06	
\glsresetsubentrycounter: bug fix: move conditional inside command	228	\@gls@@automake@immediate: new	191
\glsstepentry: bug fix: move conditional inside command	228	\@gls@automake@immediate: new	190
\glsstepsubentry: bug fix: move conditional inside command	228	altlong4col-booktabs: removed superfluous \glspatchLToutput	286
\glssubentrycounterlabel: bug fix: move conditional inside command	229		

automake: switch from boolean to choice	36	\@GLSuserv@: added accessibility support	358
\gls@automake@nr: new	36	\@GLSuservi@: added accessibility support	358
\glsfielddedef: changed from \edef to \protected@csedef	91	\@GLSdesc@: added accessibility support	355
\glsfieldxdef: changed from \edef to \protected@csxdef	91	\@GLSdescplural@: added accessibility support	355
\ifglsautomake: now defined explicitly instead of through boolean key	36	\@GLSfirst@: added accessibility support	354
noglossaryindex: new	41	\@GLSfirstplural@: added accessibility support	354
4.43 – 2019-09-28		\@GLSname@: added accessibility support	355
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01	27	\@GLSplural@: added accessibility support	354
4.44 – 2019-12-06		\@GLssymbol@: added accessibility support	356
\glsprefix@record@hook: new	328	\@GLssymbolplural@: added accessibility support	356
4.45 – 2020-02-13		\@GLStext@: added accessibility support	353
General: removed backward compatibility use of symbol key	336	\@GLSuseri@: added accessibility support	356
\@do@write@glslabels: new	35	\@GLSuserii@: added accessibility support	357
\@glsshowtarget: new	8	\@GLSuseriii@: added accessibility support	357
\@GLSdesc@: added accessibility support	355	\@GLSuseriv@: added accessibility support	357
\@GLSdescplural@: added accessibility support	355	\@GLSuserv@: added accessibility support	358
\@GLSfirst@: added accessibility support	354	\@GLSuservi@: added accessibility support	358
\@GLSfirstplural@: added accessibility support	354	\@do@write@glslabels: new	35
\@GLSname@: added accessibility support	355	\@domakeglossaries: new	34
\@GLSplural@: added accessibility support	354	\@gls@access@display: deprecated	342
\@GLSsymbol@: added accessibility support	356	\@gls@fieldaccess@display: new	342
\@GLSsymbolplural@: added accessibility support	356	\@glsdesc@: added accessibility support	355
\@GLStext@: added accessibility support	353	\@glsdescplural@: added accessibility support	355
\@GLSuseri@: added accessibility support	356	\@glsfirst@: added accessibility support	354
\@GLSuserii@: added accessibility support	357	\@glsfirstplural@: added accessibility support	354
\@GLSuseriii@: added accessibility support	357	\@glsname@: added accessibility support	355
\@GLSuseriv@: added accessibility support	357		

\@glsplural@: added accessibility support	354	\Glsentryprefixfirstplural: added \glsdetoklabel	327
\@glsshowaccsupp: new	9	\glentryprefixfirstplural: added \glsdetoklabel	326
\@glssymbol@: added accessibility support	356	\Glsentryprefixplural: added \glsdetoklabel	327
\@glssymbolplural@: added accessibility support	356	\glentryprefixplural: added \glsdetoklabel	326
\@glstext@: added accessibility support	353	\glentrytitlecase: added existence check	169
\@glseri@: added accessibility support	356	\glentryuseriaccess: new	340
\@glserii@: added accessibility support	357	\glentryuseriiaccess: new	340
\@glseriii@: added accessibility support	357	\glentryuseriiiaccess: new	340
\@glseriv@: added accessibility support	357	\glentryuserivaccess: new	340
\@glserv@: added accessibility support	358	\glentryuserviaccess: new	340
\@glservi@: added accessibility support	358	\glsfieldaccsupp: new	341
\@org@Gls@acentryname: added check for \glsshortaccessdisplay	166	\glsfirstaccessdisplay: added check for existence	343
acronyms: changed \renewcommand to \def	20	\glsfirstpluralaccessdisplay: added check for existence	343
debug: showaccsupp	7	\glslongaccessdisplay: added check for existence	344
disablemakegloss: new	34	\glslongpluralaccessdisplay: added check for existence	344
\gls@accessibility: new	333	\glsnameaccessdisplay: added check for existence	342
\gls@accsupp@engine: new	333	\glspluralaccessdisplay: added check for existence	343
\glsaccessibility: new	333	\glsprefixsep: new	328
\glsdefaultshortaccess: new	369	\glssee: switched to \newrobustcmd	209
\glsdescriptionaccessdisplay: added check for existence	343	\glsseeformat: switched to \newrobustcmd	209
\glsdescriptionpluralaccessdisplay: added check for existence	344	\glsseeitem: switched to \newrobustcmd	210
\glentrydescpluralaccess: corrected field reference	339	\glsseelist: switched to \newrobustcmd	209
\glentryfirstpluralaccess: switched to using \@gls@entry@field	339	\glsshortaccessdisplay: added check for existence	344
\glentryparent: new	170	\glsshortaccsupp: new	342
\Glsentryprefix: added \glsdetoklabel	327	\glsshortplaccsupp: new	342
\glentryprefix: added \glsdetoklabel	326	\glsshortpluralaccessdisplay: added check for existence	344
\Glsentryprefixfirst: added \glsdetoklabel	326	\glsshowaccsupp: new	9
\glentryprefixfirst: added \glsdetoklabel	326	\glsshowtargetfont: new	8
		\glsshowtargetouter: new	8
		\glsshowtargetsymbols: new	8
		\glssymbolaccessdisplay: added check for existence	343

<code>\glssymbolpluralaccessdisplay:</code>		<code>\ifglossaryexists:</code> added	
added check for existence	343	starred form	64
<code>\glstextaccessdisplay:</code> added		<code>\s@ifglossaryexists:</code> new	64
check for existence	343	<code>\setglossarypreamble:</code> switched	
<code>\glsuseriaccessdisplay:</code> new	344	to starred form of	
<code>\glsuseriiaccessdisplay:</code> new	344	<code>\ifglossaryexists</code>	51
<code>\glsuseriiiaccessdisplay:</code>		4.47 – 2021-09-20	
new	345	General: changed <code>\x</code> to <code>\@gls@x</code>	237
<code>\glsuserivaccessdisplay:</code> new	345	<code>\@do@esc@wrglossary:</code> changed	
<code>\glsuservaccessdisplay:</code> new	345	<code>\edef</code> to <code>\protected@edef</code>	206
<code>\glsuserviiaccessdisplay:</code> new	345	<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>	204
<code>\ifglshaslong:</code> made robust	68	<code>\@do@write@glslabels:</code> fully	
<code>\ifglshasprefix:</code> added		expand list	35
<code>\glsdetoklabel</code>	327	<code>\@gls@@default@acr@descplural:</code>	
<code>\ifglshasprefixfirst:</code> added		new	259
<code>\glsdetoklabel</code>	327	<code>\@gls@@default@acr@firstplural:</code>	
<code>\ifglshasprefixfirstplural:</code>		new	259
added <code>\glsdetoklabel</code>	328	<code>\@gls@@default@acr@plural:</code>	
<code>\ifglshasprefixplural:</code> added		new	259
<code>\glsdetoklabel</code>	327	<code>\@gls@noexpand@assign@default@plurals:</code>	
<code>\ifglshasshort:</code> made robust	68	new	258
<code>\ifglshassymbol:</code> made robust	68	<code>\@glo@do@sortentries:</code> changed	
<code>\makeglossaries:</code> let		<code>\edef</code> to <code>\protected@edef</code>	216
<code>\@makeglossary</code> to <code>\@gobble</code>		<code>\@glo@sortentries:</code> changed	
instead of <code>\relax</code>	193	<code>\edef</code> to <code>\protected@edef</code>	215
<code>restoremakeloss:</code> new	34	<code>\@glo@storeentry:</code> changed	
<code>user1access:</code> new	335	<code>\edef</code> to <code>\protected@edef</code>	102
<code>user2access:</code> new	335	<code>\@gls@checkesc:</code> new	131
<code>user3access:</code> new	335	<code>\@gls@deprecated@acr@warning:</code>	
<code>user4access:</code> new	336	new	261
<code>user5access:</code> new	336	<code>\@gls@link:</code> changed <code>\edef</code> to	
<code>user6access:</code> new	336	<code>\protected@edef</code>	122
<code>writelabels:</code> new	36	<code>\@gls@setupsort@use:</code> changed	
<code>\xglfieldaccsupp:</code> new	341	<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	211	<code>\@makeglossaries@warn@noprintglossary:</code>	
<code>\@printgloss@checkexists@allowignored:</code>		new	194
new	211	<code>\@write@glslabelsname:</code> new	35
<code>\@printgloss@checkexists@noignored:</code>		<code>\do@write@glslabelsname:</code> new	35
new	211	<code>\forlentries:</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>	213	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	259
<code>\gls@glsdefs@addlabel</code> : new ..	84	<code>\SetDUASStyle</code> : added deprecated	
<code>\gls@glsdefs@labellist</code> : new ..	84	warnings	261
<code>\gls@save@numberlist</code> : changed		<code>\SetFootnoteAcronymStyle</code> :	
<code>\edef</code> to <code>\protected@edef</code>	210	added deprecated warnings	260
<code>\glsadd</code> : changed <code>\edef</code> to		<code>\SetSmallAcronymStyle</code> : added	
<code>\protected@edef</code>	175	deprecated warnings	261
<code>\glsaddall</code> : changed <code>\edef</code> to		<code>writeglslabelnames</code> : new	36
<code>\protected@edef</code>	176	4.48 – 2021-10-19	
<code>\glsdefs@newdocentry</code> : new ..	85	<code>\glscapitalisewords</code> : new ...	169
<code>\glsdisplaynumberlist</code> : changed		<code>\glslistexpandedname</code> : new ..	274
<code>\edef</code> to <code>\protected@edef</code>	174	<code>\glslistinit</code> : new	274
<code>\glsfieldddef</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>	274
<code>\glsfieldfetch</code> : changed <code>\edef</code>		4.50 – 2022-10-14	
to <code>\protected@edef</code>	92	General: added initialisation ...	239
<code>\glsfieldgdef</code> : changed <code>\edef</code> to		allow markup to be stripped	
<code>\protected@edef</code>	91	from location	237
<code>\glsfieldxdef</code> : changed <code>\edef</code> to		corrected description mapping	336
<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>	269	switched to using	
<code>\GlsSetQuote</code> : added check for		<code>\glsuppercase</code>	
<code>backslash</code>	186 141–155, 157, 159, 161, 163	
<code>\glsstepsubentry</code> : changed		<code>\@do@esc@wrglossary</code> : moved	
<code>\edef</code> to <code>\protected@edef</code>	228	<code>\romannumeral</code> inside allow	
<code>\ifglsfieldcseq</code> : changed <code>\edef</code>		primitive mods	206
to <code>\protected@edef</code>	94	<code>\@gls@@default@acr@descplural</code> :	
<code>\ifglshaschildren</code> : changed		removed	259
<code>\edef</code> to <code>\protected@edef</code> ..	67	<code>\@gls@@default@acr@firstplural</code> :	
<code>\ifignoredglossary</code> : changed		removed	259
<code>\edef</code> to <code>\protected@edef</code> ..	75	<code>\@gls@@default@acr@plural</code> :	
<code>\makenoidxglossaries</code> : changed		removed	259
<code>\edef</code> to <code>\protected@edef</code>	195	<code>\@gls@default@entryfmt</code> :	
<code>\new@glossaryentry</code> : allow		removed	345
overwrite if defined in		<code>\@gls@navhypertarget</code> : new ..	269
<code>.glsdefs</code>	84	<code>\@gls@noexpand@assign@default@plurals</code> :	
<code>\newignoredglossary</code> : changed		removed	258
<code>\edef</code> to <code>\protected@edef</code> ..	74	<code>\Gls@acentryname</code> : removed	
<code>\ns@newglossary</code> : changed <code>\edef</code>		<code>mfirstuc</code> internal commands	165
to <code>\protected@edef</code>	72	<code>\Glsdisp</code> : new	140
<code>\SetDescriptionDUAAcronymStyle</code> :		<code>\Glsentryfield</code> : new	164
added deprecated warning ..	259		

\@domakeglossaries: added		
check for \nofiles	34	
\@gls@automake: added check for		
void codepage	189	
\@gls@automake@immediate:		
added check for void		
codepage	190	
\@gls@automake@types: new	191	
\@gls@defaultttype: new	71	
\@gls@do@automake@err: new	36	
\@gls@do@glsunset: new	119	
\@gls@entry@field: switched to		
\csuse	163	
\@gls@escbsdq: check		
esclocations setting	125	
\@gls@getcounterprefix:		
changed \edef to \def	207	
\@gls@getgrouptitle: switched		
to L ^A T _E X3 syntax	233	
\@gls@glslink@hyper@update@hook:		
new	118	
\@gls@hypergroup: updated to		
use etoolbox	270	
\@gls@noist@toolate: new	188	
\@gls@removedoption: new	27	
\@gls@requested@languages:		
new	31	
\@gls@restore@glslocal: new	119	
\@gls@sanitizedesc: removed	24	
\@gls@save@glslocal: new	119	
\@gls@setupsort@clear: new	18	
\@gls@defaultplural: Removed	80	
\@org@Gls@acrentryname: saved		
original definition of		
\@Gls@acrentryname	166	
\@xp@gls@getcounterprefix:		
new	207	
\@xp@gls@hypernumber: new	236	
automake: added makegloss and		
lite options	36	
automakegloss: new	37	
automakeglosslite: new	37	
compatible-2.07: deprecated	38	
compatible-3.07: deprecated	38	
\compatibleglossentry:		
removed	230, 333	
\compatiblesubglossentry:		
removed	231, 333	
\CustomAcronymFields:		
removed	261	
\CustomNewAcronymDef:		
removed	261	
\DefaultNewAcronymDef:		
removed	258, 369	
\defglsdisplay: removed	118	
\defglsdisplayfirst: removed	118	
\DescriptionDUANewAcronymDef:		
removed	259	
\DescriptionFootnoteNewAcronymDef:		
removed	259, 369	
\DescriptionNewAcronymDef:		
removed	260, 369	
dua: switched to using		
\glsupercase	252, 253	
dua-desc: removed \acronymfont		
from around long form	254	
\DUANewAcronymDef: removed	261	
esclocations: default now false	12	
\findrootlanguage: removed	62	
footnote: switched to using		
\glsupercase	255	
\FootnoteNewAcronymDef:		
removed	260, 369	
\glossary: removed	39	
\glossaryentryfield: removed	231	
\glossarymark: switched to using		
\glsupercase	52	
\glossarystyle: removed	234	
\glossarysubentryfield:		
removed	231	
\gls@automake@makegloss: new	36	
\gls@automake@option: new	36	
\gls@codepage: set default		
codepage to utf8 if		
\inputencodingname not		
defined and switched to		
\ifdefvoid	33	
\gls@delimN: renamed \@delimN		
to \gls@delimN	236	
\gls@delimR: renamed \@delimR		
to \gls@delimR	236	
\gls@disablepagerefexpansion:		
added check for esclocations	201	
\gls@noidx@nonumberlist: new	78	
\gls@noidxglossary: added		
\glswrglossdisablelocationcmds		
	198	
\gls@xdy@set@lang: new	33	
\glsaddkey: allow expansion in		
PDF bookmarks	89	

switched to using		
\gluppercase	90	
\GlsAddXdyAttribute: remove		
duplicate counterless		
attribute definitions	56	
\Glsdisp: new	140	
\glsdisplay: removed	118	
\glsdisplayfirst: removed	117	
\glsdoparenifnotempty:		
replaced		
\DeclareRobustCommand with		
\newrobustcmd	260	
\glsdoshowtarget: new	8	
\glsencapwrcontent: new	123	
\Glsentrydesc: allow expansion		
in PDF bookmarks	167	
\Glsentrydescplural: allow		
expansion in PDF		
bookmarks	167	
\Glsentryfirst: allow expansion		
in PDF bookmarks	168	
\Glsentryfirstplural: allow		
expansion in PDF		
bookmarks	169	
\glsentryfmt: obsolete		
commands replaced with		
\glsgenentryfmt	112	
\Glsentrylong: allow expansion		
in PDF bookmarks	172	
\Glsentrylongpl: allow		
expansion in PDF		
bookmarks	172	
\Glsentryname: allow expansion		
in PDF bookmarks	165	
\Glsentryplural: allow		
expansion in PDF		
bookmarks	167	
\Glsentryprefix: allow		
expansion in PDF		
bookmarks	327	
\Glsentryprefixfirst: allow		
expansion in PDF		
bookmarks	326	
\Glsentryprefixfirstplural:		
allow expansion in PDF		
bookmarks	327	
\Glsentryprefixplural: allow		
expansion in PDF		
bookmarks	327	
\Glsentryshort: allow expansion		
in PDF bookmarks	172	
\Glsentryshortpl: allow		
expansion in PDF		
bookmarks	172	
\Glsentrysymbol: allow		
expansion in PDF		
bookmarks	168	
\Glsentrysymbolplural: allow		
expansion in PDF		
bookmarks	168	
\Glsentrytext: allow expansion		
in PDF bookmarks	167	
\glsentrytitlecase: use		
sentence case for PDF		
bookmark	169	
\Glsentryuseri: allow expansion		
in PDF bookmarks	170	
\Glsentryuserii: allow		
expansion in PDF		
bookmarks	170	
\Glsentryuseriii: allow		
expansion in PDF		
bookmarks	170	
\Glsentryuseriv: allow		
expansion in PDF		
bookmarks	171	
\Glsentryuserv: allow expansion		
in PDF bookmarks	171	
\Glsentryuservi: allow		
expansion in PDF		
bookmarks	171	
\glsfielddef: made robust	92	
\glsfieldedef: made robust	91	
\glsfieldfetch: made robust	92	
\glsfieldgdef: made robust	91	
\glsfieldxdef: made robust	91	
\glsgenacfmt: switched to using		
\gluppercase	115	
\glsgenentryfmt: switched to		
using \gluppercase	113	
\glsgetgrouplabel: converted to		
use L ^A T _E X3	233	
\glsgetgrouptitle: made		
robust	233	
\gls glossarymark: switched to		
\gluppercase	51	
\glsifhyper: removed	119	
\glsindexingsetting: new	15	
\glsinlineifhaschildren: new	272	

\Glslink: new	121	\istfile: removed	198
\glslocationcstoencap: new	50	kernelglossredefs: corrected	
\glslowercase: new	4	names of choice variables	39
\glsmfuaddmap: new	5	languages: new	31
\glsmfublocker: new	5	\makeglossary: removed	38
\glsmfuexcl: new	5	mfirstuc: new	41
\glsnavhyperlink: changed		\SetAcronymStyle: removed	261
\@glo@type to		\SetCustomDisplayStyle:	
\currentglossary	268	removed	261
\glsnavhypertarget: changed		\SetCustomStyle: removed	261
\@glo@type to		\SetDefaultAcronymStyle:	
\currentglossary	269	removed	259
\glsnavigation: changed		\SetDescriptionAcronymDisplayStyle:	
\@glo@type to		removed	260
\currentglossary	270	\SetDescriptionAcronymStyle:	
switched to \letcs	270	removed	260
\glsnoidxprenumberlist: new	78	\SetDescriptionDUAAcronymDisplayStyle:	
\glsnumberlistloop: corrected		removed	259
parameter number	197	\SetDescriptionDUAAcronymStyle:	
\glsseeitemformat: added test		removed	259
for short field	210	\SetDescriptionFootnoteAcronymDisplayStyle:	
\glssentencecase: new	4	removed	259
\glsshowtargetfonttext: new	8	\SetDescriptionFootnoteAcronymStyle:	
\glsshowtargetinner: new	8	removed	259
\glstexorpdfstring: new	164	\SetDUADisplayStyle: removed	261
\glssupercase: new	4	\SetDUASyle: removed	261
\GlsWarnAddProtectedPageFmt:		\SetFootnoteAcronymDisplayStyle:	
new	202	removed	260
\glswrglossdisableanchorcmds:		\SetFootnoteAcronymStyle:	
new	203	removed	260
\glswrglossdisablelocationcmds:		\SetGenericNewAcronym:	
new	201	corrected long plural suffix	245
\glswrglosslocationtarget:		switched to using	
new	203	\glssupercase	245
\glswrglosslocationtextfmt:		\setglossentrycompatibility:	
new	203	removed	231
\ifgl\$fieldcseq: made robust	94	\SetSmallAcronymDisplayStyle:	
\ifgl\$fielddefeq: made robust	93	removed	260
\ifgl\$fieldeq: made robust	93	\SetSmallAcronymStyle:	
\ifgl\$fieldvoid: new	68	removed	261
\ifgl\$resetcurrcount: new	106	\SmallNewAcronymDef:	
\ifgl\$swallowprimitivemods:		removed	260, 369
switch default to false	204	sort: added clear	15

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

Symbols	
\!	126, 129, 186, 446, 447, 695, 697, 698, 728
\"	26, 126–129, 132, 383, 444–446, 448, 633, 695–698, 700
\#	131, 181, 471, 472, 699, 700, 723
\%	177, 183, 469, 473, 574–576, 721, 725, 827–829
\&	45, 174, 394, 468, 644, 720
\'	26, 383, 633
\.	13, 26, 376, 383, 626, 633
\=	26, 383, 633
\?	126, 128, 186, 444–446, 475, 695–697, 727
\@	85, 416, 667
\@@delimN	506, 758
\@do@@wrglossary	195, 204, 206, 206, 481, 487, 488, 733, 738, 739
\@do@esc@wrglossary	203, <u>204</u> , 486, 487, 738, 739
\@do@noesc@wrglossary	203, <u>203</u> , 486, 738
\@do@wrglossary	175, 201, <u>203</u> , 468, 485, 486, 720, 737, 738
\@do@write@glslabels	35, <u>35</u> , 389, 639
\@glo@assign@sortkey	195, <u>227</u> , 482, 500, 734, 752
\@glo@list	64, 405, 655
\@glo@sort	26, 383, 633
\@glo@type	212, 490, 742
\@glossarysec	10, 10, 52, 53, 375, 398, 624, 625, 648
\@glossaryseclabel	10, <u>10</u> , 11, 53, 225, 375, 398, 500, 625, 648, 751, 752
\@glossarysecstar	10, <u>10</u> , 11, 52, 53, 225, 375, 398, 500, 625, 648, 751, 752
\@gls@@default@acr@descplural	<u>259</u> , 772
\@gls@@default@acr@firstplural	<u>259</u> , 772
\@gls@@default@acr@plural	<u>259</u> , 772
\@gls@checkactual	130, 131, 448, 699
\@gls@checkbar	129, 130, 447, 698
\@gls@checkescactual	128, 445, 446, 696, 697
\@gls@checkescbar	128, 129, 446, 697
\@gls@checkesclevel	129, 446, 447, 697, 698
\@gls@checkescquote	127, 128, 187, 445, 476, 477, 696, 728, 729
\@gls@checklevel	130, 447, 448, 698, 699
\@gls@checkquote	127, 185, 444, 445, 475, 695, 696, 726, 727
\@gls@codepage	189, 190
\@gls@default@entryfmt	<u>345</u> , 433, 439, 440, 544, 684, 690, 797
\@gls@expand@field	24, 82, 82, 88, 382, 414, 415, 418, 520–523, 525, 527, 529, 531, 565–569, 632, 665, 668, 669, 772, 774, 775, 777, 779, 780, 782, 784, 819–822
\@gls@extramakeindexopts	185, 192, 474, 480, 726, 732
\@gls@fixbraces	209, <u>209</u> , 489, 741
\@gls@navhypertarget	269, <u>269</u>
\@gls@noexpand@assign@default@plurals	<u>258</u> , 772, 819
\@gls@noexpand@field	24, 81, 82, 382, 414, 632, 664, 665
\@gls@noidx@no@sanitizesort	26, 383, 633
\@gls@noidx@nosanitizesort	<u>26</u> , 197, 383, 483, 633, 735
\@gls@noidx@sanitizesort	<u>25</u> , 383, 632
\@gls@nosanitizesort	25, <u>25</u> , 197, 382, 383, 483, 632, 735
\@gls@sanitizesort	25, <u>25</u> , 197, 382, 383, 483, 632, 735
\@gls@xdycheckbackslash	132, 133, 449, 700, 701
\@gls@xdycheckquote	132, 448, 700

\@glslocalreset	105, 105, 107, 429, 431, 680, 681	\@GLSsymbol	148, 456, 708
\@glslocalunset	105, 105, 107, 429, 430, 680, 681	\@GLSsymbol@	148, 356, 456, 554, 708, 807
\@glsreset	105, 105, 107, 429, 430, 680, 681	\@GLSsymbolplural	149, 456, 708
\@glsreset@	105, 105, 107, 429, 430, 680, 681	\@GLSsymbolplural@	149, 356, 456, 554, 708, 808
\@glsshowtarget	7, 8, 9, 373, 374, 623, 624	\@GLStext	141, 453, 705
\@glsunset	105, 105, 107, 429, 430, 680, 681	\@GLStext@	141, 353, 453, 553, 705, 806
\@newglossaryentry@defcounters	106, 107, 430, 681	\@GLSuseri	150, 457, 708
\@this@glo@	64, 405, 655	\@GLSuseri@	150, 356, 457, 555, 708, 808
\@ACRfull	242, 508, 760	\@GLSuserii	151, 457, 709
\@ACRfullpl	243, 508, 509, 760	\@GLSuserii@	151, 357, 457, 555, 709, 808
\@ACRlong	161, 241, 353, 462, 508, 552, 713, 760, 805	\@GLSuseriii	152, 457, 709
\@ACRlongpl	163, 241, 463, 509, 714, 761	\@GLSuseriii@	152, 357, 457, 555, 709, 808
\@ACRshort	157, 241, 352, 459, 508, 551, 711, 760, 805	\@GLSuseriv	153, 458, 709
\@ACRshortpl	159, 241, 460, 461, 509, 712, 761	\@GLSuseriv@	153, 357, 458, 555, 709, 808
\@Acrfull	242, 508, 760	\@GLSuseriv	154, 458, 710
\@Acrfullpl	243, 508, 760	\@GLSuseriv@	154, 358, 458, 556, 710, 809
\@Acrlong	160, 241, 352, 461, 508, 552, 713, 760, 805	\@GLSuservi	155, 458, 710
\@Acrlongpl	162, 241, 462, 508, 714, 760	\@GLSuservi@	155, 358, 458, 556, 710, 809
\@Acrshort	156, 351, 459, 551, 711, 804	\@Gls	135, 450, 702
\@Acrshortpl	158, 460, 712	\@Gls@	108, 110, 135, 135, 330, 431, 432, 450, 590, 682, 683, 702, 843
\@Alph	202, 205, 206, 486, 487, 738, 739	\@Gls@@link	121
\@GLS	136, 451, 702, 703	\@Gls@acrentryname	165, 244, 464, 509, 715, 761
\@GLS@	136, 136, 331, 451, 591, 703, 844	\@Gls@entry@field	164, 164, 165, 166, 419, 463–467, 669, 715–718
\@GLSdesc	146, 455, 707	\@Gls@entryname	165, 165, 244, 464, 509, 715, 761
\@GLSdesc@	146, 355, 455, 554, 707, 807	\@GlsSetXdyFirstLetterAfterDigits	177, 178, 469, 721
\@GLSdescplural	147, 456, 707	\@GlsSetXdyNumberGroupOrder	178, 469, 470, 721
\@GLSdescplural@	147, 355, 456, 554, 707, 807	\@Glsdesc	146, 455, 707
\@GLSfirst	142, 454, 705	\@Glsdesc@	146, 355, 455, 554, 707, 807
\@GLSfirst@	142, 354, 454, 553, 705, 806	\@Glsdescplural	147, 455, 707
\@GLSfirstplural	144, 454, 706	\@Glsdescplural@	147, 355, 455, 554, 707, 807
\@GLSfirstplural@	144, 354, 455, 553, 706, 807	\@Glsdisp	140, 140
\@GLSname	145, 455, 706	\@Glsentryfield	89, 164, 167–172, 326, 327
\@GLSname@	145, 355, 455, 554, 707, 807	\@Glsfirst	142, 454, 705
\@GLSpl	138, 139, 452, 704	\@Glsfirst@	142, 354, 454, 553, 705, 806
\@GLSpl@	138, 139, 332, 452, 592, 704, 845	\@Glsfirstplural	144, 454, 706
\@GLSplural	143, 454, 706	\@Glsfirstplural@	144, 354, 454, 706
\@GLSplural@	143, 354, 454, 553, 706, 806		

<code>\@Glsname</code>	145, 455, 706	<code>\@addtoacronymlists</code>	21
<code>\@Glsname@</code>	145, 355, 455, 553, 706, 807	<code>\@afterheading</code>	276, 578, 595, 831, 849
<code>\@Glspl</code>	137, 138, 452, 703	<code>\@alph</code>	201, 205, 206, 486, 487, 738, 739
<code>\@Glspl@</code>	108,	<code>\@arabic</code>	
	111, 138, 138, 331, 431, 433,		202, 205, 206, 486, 487, 738, 739
	452, 590, 591, 682, 683, 703, 844	<code>\@auxout</code>	71, 73, 109, 192,
<code>\@Glsplural</code>	143, 454, 706		193, 195, 198, 210, 214, 215,
<code>\@Glsplural@</code>			269, 409, 432, 480–482, 484,
 143, 354, 454, 553, 706, 806		490, 492, 592, 659, 660, 682,
<code>\@Glsymbol</code>	148, 456, 707		683, 732–734, 736, 742, 744, 845
<code>\@Glsymbol@</code>		<code>\@backslashchar</code>	125, 126, 132, 186,
 148, 356, 456, 554, 707, 807		443, 449, 694, 695, 700, 701, 728
<code>\@Glsymbolplural</code>	149, 456, 708	<code>\@bsphack</code>	200, 485, 737
<code>\@Glsymbolplural@</code>		<code>\@cGls</code>	110, 110, 432, 683
 149, 356, 456, 554, 708, 808	<code>\@cGls@</code>	108, 110, 110, 431, 432, 682, 683
<code>\@Glstext</code>	141, 453, 705	<code>\@cGlspl</code>	111, 432, 683
<code>\@Glstext@</code>	141, 353, 453, 553, 705, 806	<code>\@cGlspl@</code>	
<code>\@Glsuser@i</code>	356–358, 555, 556, 808, 809		108, 111, 111, 431–433, 682, 683
<code>\@Glsuseri</code>	150, 457, 708	<code>\@cclv</code>	288, 601, 602, 855
<code>\@Glsuseri@</code>	150, 356, 457, 708	<code>\@cgls</code>	109, 109, 432, 683
<code>\@Glsuserii</code>	151, 457, 709	<code>\@cgls@</code>	108, 109, 110, 431, 432, 681, 683
<code>\@Glsuserii@</code>	151, 357, 457, 709	<code>\@cglspl</code>	110, 111, 432, 683
<code>\@Glsuseriii</code>	152, 457, 709	<code>\@cglspl@</code>	
<code>\@Glsuseriii@</code>	152, 357, 457, 709		108, 111, 111, 431, 432, 682, 683
<code>\@Glsuseriv</code>	153, 458, 709	<code>\@chapter</code>	43, 393, 643
<code>\@Glsuseriv@</code>	153, 357, 458, 709	<code>\@classoptionslist</code>	41, 392, 642
<code>\@Glsuserv</code>	154, 458, 710	<code>\@closegls</code>	189, 189, 477, 478, 729, 730
<code>\@Glsuserv@</code>	154, 358, 458, 710	<code>\@colht</code>	288, 602, 855
<code>\@Glsuservi</code>	155, 458, 710	<code>\@colroom</code>	288, 602, 855
<code>\@Glsuservi@</code>	155, 358, 458, 710	<code>\@currentlabelname</code>	
<code>\@Mi</code>	288, 601, 855	 11, 226, 375, 500, 625, 752
<code>\@PGLS</code>	331, 331, 591, 844	<code>\@curroptions</code>	41, 392, 642
<code>\@PGLS@</code>	331, 331, 591, 844	<code>\@declaredoptions</code>	41, 392, 642
<code>\@PGLSpl</code>	331, 332, 591, 844	<code>\@delimN</code>	506, 758
<code>\@PGLSpl@</code>	332, 332, 591, 844	<code>\@delimR</code>	506, 758
<code>\@Pgls</code>	329, 329, 590, 843	<code>\@disable@onlypremakeg</code>	
<code>\@Pgls@</code>	329, 329, 590, 843	 43, 193, 393, 480, 644, 732
<code>\@Pglspl</code>	330, 330, 590, 843	<code>\@disable@premakecs</code>	43, 43, 394, 644
<code>\@Pglspl@</code>	330, 330, 590, 843	<code>\@disabled@glsaddxdycounters</code>	
<code>\@Roman</code>	202, 205, 206, 486, 487, 738, 739	 55, 57, 400, 650, 651
<code>\@acrfull</code>	242, 242, 507, 759	<code>\@do@addcounter</code>	55, 399, 400, 650
<code>\@acrfullpl</code>	243, 508, 760	<code>\@do@auxoutstuff</code>	214, 215, 492, 744
<code>\@acrlong</code>	159, 241, 352,	<code>\@do@glossentry</code>	502, 536, 754, 789
	461, 507, 552, 712, 713, 759, 805	<code>\@do@gls@checkesc</code>	126, 695
<code>\@acrlongpl</code>	161, 241, 462, 508, 714, 760	<code>\@do@gls@getcounterprefix</code>	
<code>\@acrshort</code>	155, 156, 241, 351, 459,	 207, 486, 487, 738, 739
	507, 508, 551, 710, 759, 760, 804	<code>\@do@gls@islistofacronyms</code>	
<code>\@acrshortpl</code> 21, 380, 630
 157, 241, 460, 508, 711, 760	<code>\@do@glssee</code>	100, 426, 677
<code>\@addtoacronymlists</code>	21, 380, 630	<code>\@do@ifinlist</code>	55, 399, 649

<code>\@do@newglossaryentry</code>	<code>\@for</code>
. . . . 244, 245, 509, 519–532,	45, 55, 56, 63, 64, 86, 87, 125,
565–569, 761, 771–786, 818–822	179–181, 191, 193, 201, 209,
<code>\@do@seeglossary</code> 195,	215, 216, 246, 269, 270, 387,
208, 209, 481, 489, 733, 740, 741	392, 393, 399, 400, 405, 416,
<code>\@do@subglossentry</code> 503, 536, 755, 789	417, 443, 470, 471, 480, 481,
<code>\@do@wrglossary</code> 123,	486, 489, 493, 510, 520, 522,
201, 442, 485, 572, 692, 737, 825	524, 525, 527, 530, 531, 533,
<code>\@do@write@glslabels</code> 35, 36, 389, 639	572, 592, 593, 637, 642, 644,
<code>\@do@writeaux@info</code> 210, 490, 742	650, 655, 667, 668, 694, 722,
<code>\@domakeglossaries</code>	723, 732, 733, 738, 741, 744,
. . . 34, 34, 35, 192, 194, 388,	745, 762, 772, 775, 776, 778,
389, 480, 481, 638, 639, 732, 733	780, 783, 784, 786, 826, 845, 846
<code>\@ehc</code> 288, 601, 855	<code>\@glo@@desc</code> 99, 425, 676
<code>\@empty</code> 14, 18,	<code>\@glo@@symbol</code> 99, 426, 677
21, 38, 40, 41, 43, 55, 56, 60,	<code>\@glo@access</code> . . 334, 337, 342, 536,
62, 63, 97, 102, 123, 124, 135–	538, 539, 541, 542, 790–792, 795
139, 156–163, 179, 182, 184,	<code>\@glo@addchildren</code>
189–191, 199, 200, 208, 234,	216, 216, 220, 493, 496, 745, 748
351–353, 377, 379, 380, 390,	<code>\@glo@assign@sortkey</code> . . 193, 195,
392, 393, 399, 400, 402, 404,	226, 481, 482, 500, 732, 734, 752
405, 424, 427, 442, 450–452,	<code>\@glo@autosee</code> 100, 100, 426, 677
459–463, 470, 472, 474, 478,	<code>\@glo@autoseehook</code> 100, 101, 426, 677
479, 484, 485, 488, 504, 520,	<code>\@glo@check@mkidxrangear</code>
522–529, 531, 533, 551, 552, 124, 125,
572, 574–576, 626, 628, 630,	207, 443, 488, 572, 694, 739, 825
640, 642, 643, 649, 650, 653,	<code>\@glo@check@sortallowed</code>
655, 674, 675, 678, 693, 702–	16, 16, 17–19, 194, 197, 377–
704, 710–715, 722, 724, 726,	379, 481, 483, 627–629, 733, 735
729–731, 736, 737, 740, 756,	<code>\@glo@childlist</code> 216, 493, 745
772, 774, 776, 778, 780–782,	<code>\@glo@counter</code> 77,
784, 786, 804–806, 826, 828, 829	95, 99, 412, 423, 425, 662, 674, 676
<code>\@end@fixbraces</code> 209, 489, 741	<code>\@glo@counterprefix</code> 198, 203, 206–
<code>\@end@fortrue</code> . 31, 67, 87, 269, 387,	208, 234, 237, 484, 486–488,
407, 417, 592, 637, 657, 668, 845	504, 506, 736, 738–740, 756, 758
<code>\@esphack</code> 201, 485, 737	<code>\@glo@default@sorttype</code>
<code>\@expandtwoargs</code> 41, 392, 642	. . 15, 15, 196, 218, 219, 377,
<code>\@firstofone</code> 26, 35,	482, 495, 496, 627, 734, 746–748
274, 383, 384, 389, 633, 639, 848	<code>\@glo@defaultcounter</code> 98, 99, 425, 676
<code>\@firstofthree</code> 120, 134,	<code>\@glo@desc</code> 75, 94, 95, 97, 99,
135, 137, 139, 156, 158, 159,	411, 422–426, 661, 673, 674, 676
161, 351–353, 440, 450, 451,	<code>\@glo@descaccess</code>
453, 459–462, 551, 552, 691,	335, 337, 338, 537–539, 790–792
702–704, 710, 712–714, 804–806	<code>\@glo@descplural</code> 75,
<code>\@firstoftwo</code>	94, 95, 411, 422, 423, 661, 673
. . . . 29–31, 86, 87, 120, 137–	<code>\@glo@descpluralaccess</code>
139, 158, 159, 161–163, 386,	335, 337, 338, 537–539, 790–792
387, 416, 417, 440, 451, 452,	<code>\@glo@do@sortentries</code>
460–463, 635–637, 667, 668, 215, 216, 493, 744, 745
691, 703, 704, 711, 712, 714, 715	<code>\@glo@entry</code> 176, 469, 720

<code>\@glo@entryprefix</code>	<code>\@glo@plural</code> .. 76, 95, 97, 98, 411,
.... 325, 326, 587, 588, 840, 841	423–425, 567, 661, 674–676, 820
<code>\@glo@entryprefixfirst</code>	<code>\@glo@pluralaccess</code>
.... 325, 326, 587, 588, 840, 841 334, 337, 537–539, 790–792
<code>\@glo@entryprefixfirstplural</code> ..	<code>\@glo@prefix</code> . 12, 78, 96, 102, 103,
.... 325, 326, 587, 588, 840, 841	124, 125, 207, 376, 412, 423,
<code>\@glo@entryprefixplural</code>	427, 428, 443, 488, 572, 625,
.... 325, 326, 587, 588, 840, 841	662, 674, 678, 679, 694, 740, 825
<code>\@glo@esclabel</code>	<code>\@glo@range</code> ... 207, 488, 572, 740, 825
.... 102, 103, 427, 428, 678, 679	<code>\@glo@see</code>
<code>\@glo@etext</code>	100, 412, 423, 426, 662, 674, 677
433–435, 684–686	<code>\@glo@seeautonumberlist</code>
<code>\@glo@first</code> 12, <u>12</u> , 77, 376, 412, 625, 662
76,	<code>\@glo@short</code>
95, 98, 99, 411, 423, 425, 426,	68, 80, 96, 99, 407, 413, 423,
529, 568, 661, 674, 676, 782, 822	425, 568, 658, 663, 674, 676, 822
<code>\@glo@firstaccess</code>	<code>\@glo@shortaccess</code>
.... 334, 337, 537–539, 790–792 335, 337, 338, 537–539,
<code>\@glo@firstplural</code> 76, 95, 98, 411,	565–568, 790–792, 818–820, 822
423, 425, 568, 661, 674, 676, 822	<code>\@glo@shortpl</code>
<code>\@glo@firstpluralaccess</code>	80,
.... 334, 337, 537–539, 790–792	96, 99, 413, 423, 425, 519, 521,
<code>\@glo@grabfirst</code> .. 221, <u>221</u> , 497, 749	523, 525, 526, 529, 530, 565,
<code>\@glo@label</code> 78, 80, 88, 89, 91–100,	568, 663, 674, 676, 771–773,
106, 174, 175, 320, 326, 337,	775, 777, 779, 782, 783, 818, 822
338, 413, 418, 420–426, 430,	<code>\@glo@shortpluralaccess</code>
467, 468, 538–540, 588, 619,	335, 337, 338, 537–539, 790–792
620, 663, 664, 669, 671–677,	<code>\@glo@sort</code> 16, 18, 25, 26, 76, 95, 98,
681, 719, 720, 792, 793, 841, 873	102, 103, 378, 379, 383, 411,
<code>\@glo@list</code>	423, 425, 427, 428, 627, 629,
100, 426, 677	632, 633, 661, 674, 675, 678, 679
<code>\@glo@long</code> ... 68, 80, 96, 99, 407,	<code>\@glo@sortedinsert</code>
413, 423, 425, 657, 663, 674, 676	216, 217, <u>217</u> , 493, 494, 745, 746
<code>\@glo@longaccess</code>	<code>\@glo@sortentries</code> <u>215</u> , 218,
335, 337, 338, 537–539, 790–792	219, 492, 495, 496, 744, 746, 747
<code>\@glo@longpl</code>	<code>\@glo@sorthandler@case</code>
80, 96, 99, 413, 423, 425, 519, <u>218</u> , 219, 494, 495, 746, 747
521, 523, 525, 526, 529, 530,	<code>\@glo@sorthandler@letter</code>
565, 663, 674, 676, 771, 772, <u>217</u> , 218, 494, 495, 746, 747
774, 775, 777, 779, 782, 783, 818	<code>\@glo@sorthandler@nocase</code>
<code>\@glo@longpluralaccess</code> <u>218</u> , 219, 494, 496, 746, 747
335, 337, 338, 537–539, 790–792	<code>\@glo@sorthandler@word</code>
<code>\@glo@name</code> <u>217</u> , 218, 494, 495, 746
16, 75, 81,	<code>\@glo@sortinghandler</code>
95, 97–99, 378, 411, 414, 423– 215, 217, 492, 494, 744, 746
425, 627, 661, 664, 673, 675, 676	<code>\@glo@sortinglist</code> 215, 217,
<code>\@glo@no@assign@sortkey</code>	220, 492–494, 496, 744, 745, 748
.... 193, <u>227</u> , 481, 500, 732, 752	<code>\@glo@sortmacro@case</code> .. <u>219</u> , 495, 747
<code>\@glo@nonumberlist</code> . 79, 412, 413, 663	<code>\@glo@sortmacro@def</code> ... <u>220</u> , 496, 748
<code>\@glo@numfmt</code> .. 207, 488, 572, 740, 825	<code>\@glo@sortmacro@def@do</code> <u>220</u> , 496, 748
<code>\@glo@parent</code>	<code>\@glo@sortmacro@letter</code> <u>218</u> , 495, 747
18, 78, 95,	
97, 98, 102, 103, 216, 217, 379,	
412, 423, 424, 427, 428, 493,	
628, 662, 674, 675, 678, 679, 745	

<code>\@glo@sortmacro@nocase</code>	219, 495, 747	481, 535, 619, 639, 655, 659,
<code>\@glo@sortmacro@standard</code>	680, 681, 720, 732, 733, 788, 873
.....	219, 495, 747	<code>\@glo@useri</code>
<code>\@glo@sortmacro@use</code>	... 220, 496, 748	96, 99, 413, 423, 425, 663, 674, 676
<code>\@glo@sortmacro@word</code>	.. 218, 494, 746	<code>\@glo@useriaccess</code>
<code>\@glo@sorttype</code>	335, 337, 338, 537–539, 790–792
....	196, 220–222, 227, 482,	<code>\@glo@userii</code>
496, 497, 500, 734, 748, 749, 752		96, 99, 413, 423, 425, 663, 674, 676
<code>\@glo@storeentry</code> 16–18,	<code>\@glo@useriiaccess</code>
102, 377–379, 427, 627, 628, 678		335, 337, 338, 537–539, 790, 791, 793
<code>\@glo@suffi</code> 124, 125,	<code>\@glo@useriii</code>
207, 443, 488, 572, 694, 740, 825		96, 99, 413, 423, 425, 663, 674, 676
<code>\@glo@symbol</code> 68, 76, 95, 99,	<code>\@glo@useriiiaccess</code>
100, 407, 411, 423, 426, 524,		335, 337, 338, 537–539, 790, 791, 793
528, 657, 661, 674, 677, 777, 781		<code>\@glo@useriv</code>
<code>\@glo@symbolaccess</code> 334,	96, 99, 413, 423, 425, 663, 674, 676
337, 537–539, 568, 790–792, 821		<code>\@glo@userivaccess</code>
<code>\@glo@symbolplural</code> 77, 95, 336–338, 537–539, 791–793
100, 411, 423, 426, 661, 674, 677		<code>\@glo@userv</code>
<code>\@glo@symbolpluralaccess</code>	96, 99, 413, 423, 425, 663, 674, 676
334, 337, 338, 537–539, 790–792		<code>\@glo@uservaccess</code>
<code>\@glo@text</code>	.. 76, 95, 98, 99, 135–	336–338, 537, 538, 540, 791–793
140, 164–166, 411, 423, 425,		<code>\@glo@uservi</code>
426, 450–453, 463, 464, 525,		80, 96, 99, 413, 423, 425, 663, 674, 676
567, 588, 661, 674, 675, 677,		<code>\@glo@userviaccess</code>
702–704, 715, 716, 777, 820, 841		336–338, 537, 538, 540, 791–793
<code>\@glo@textaccess</code>	334, 337, 536, 538,	<code>\@glodesc</code>
539, 565–567, 790–792, 818–821		99, 426, 676
<code>\@glo@thislabel</code>	.. 101, 427, 677, 678	<code>\@glolist@</code>
<code>\@glo@thislettergrp</code>	97, 424, 675
.... 221–223, 497, 498, 749, 750		<code>\@gloname</code>
<code>\@glo@thisvalue</code> 69, 70, 408, 658	99, 425, 676
<code>\@glo@tmp</code> 71,	<code>\@glossary@default@style</code>
88, 89, 208, 418, 488, 669, 740	 11, 11, 13,
<code>\@glo@type</code> 11, 18, 77, 95–	212, 234, 262, 375, 376, 491,
100, 175, 176, 193, 198, 199,		505, 533, 625, 626, 743, 757, 786
212–215, 217, 220, 221, 225,		<code>\@glossaryentryfield</code>
226, 244, 375, 379, 411, 423–	 101, 102, 103, 427, 428, 678
426, 468, 469, 480, 484, 485,		<code>\@glossarysection</code>
491–493, 496, 497, 499, 500,	 51, 52, 397, 398, 647, 648
509, 520, 522, 523, 525, 527,		<code>\@glossarystyle</code>
529, 531, 533, 592, 593, 625,		212, 213, 225, 491, 499, 743, 751
628, 661, 673–677, 720, 732,		<code>\@glossarysubentryfield</code>
736, 737, 742–745, 748, 749,	 101, 102, 103, 427, 428, 678, 679
751, 752, 761, 772, 774, 776,		<code>\@gls</code>
778, 780, 782, 784, 786, 845, 846		134, 134, 450, 702
<code>\@glo@types</code> 35,	<code>\@gls@</code>
63, 64, 70, 72, 105, 106, 176,		108, 110, 134,
191, 193, 194, 267, 320, 389,		135, 328, 330, 431, 432, 450,
405, 409, 429, 430, 469, 480,		589, 590, 682, 683, 702, 842, 843
		<code>\@gls@automake@immediate</code>
	 191, 192, 480, 731, 732
		<code>\@gls@link</code>
		120, 120, 121, 440, 441, 691
		<code>\@gls@Hcounter</code>
		123, 124, 442, 693
		<code>\@gls@ReturnAfterFi</code>
		237, 506, 507, 758

<code>\@gls@access@display</code> .. 342 , 541 , 795	<code>\@gls@counter</code> .. 118 , 122–124 , 175 , 198 , 207 , 208 , 440–443 , 468 , 484 , 488 , 572 , 691–693 , 720 , 736 , 740 , 825
<code>\@gls@actualchar</code> .. 103 , 128 , 131 , 183 , 428 , 445 , 446 , 448 , 473 , 575 , 678 , 679 , 696 , 697 , 699 , 724 , 725 , 828	<code>\@gls@counterwithin</code> .. 14 , 15 , 377 , 626 , 627
<code>\@gls@addpredefinedattributes</code> .. 57 , 179 , 188 , 401 , 470 , 477 , 651 , 722 , 729	<code>\@gls@ctr</code> .. 55 , 399 , 650
<code>\@gls@adjustmode</code> .. 175 , 176 , 468 , 720	<code>\@gls@currentlettergroup</code> .. 221 , 223 , 496 , 498 , 748 , 750
<code>\@gls@after</code> .. 21 , 380 , 381 , 630	<code>\@gls@debugfalse</code> .. 7 , 373 , 623
<code>\@gls@assign@symbol@field</code> .. 25	<code>\@gls@debugtrue</code> .. 7 , 373 , 622 , 623
<code>\@gls@assign@symbolplural@field</code> .. 25	<code>\@gls@declareoption</code> .. 11 , 12 , 13 , 20 , 23 , 24 , 30 , 33 , 34 , 36 , 37 , 40 , 41 , 375 , 376 , 379–382 , 386 , 388 , 389 , 391 , 392 , 625 , 626 , 629–632 , 636 , 638 , 639 , 641 , 642
<code>\@gls@automake</code> .. 189 , 194 , 477 , 478 , 481 , 729 , 730 , 733	<code>\@gls@default</code> .. 112 , 433 , 684
<code>\@gls@automake@immediate</code> .. 190 , 191 , 478–480 , 730–732	<code>\@gls@default@value</code> .. 68–70 , 76 , 81 , 82 , 95 , 97–99 , 260 , 326 , 407 , 408 , 411 , 414 , 415 , 423– 426 , 528 , 587 , 588 , 657 , 658 , 661 , 664 , 665 , 673–676 , 781 , 841
<code>\@gls@automake@types</code> .. 191 , 191 , 193	<code>\@gls@defaultttype</code> .. 71 , 71
<code>\@gls@before</code> .. 21 , 380 , 630	<code>\@gls@deffile</code> .. 84–86 , 416 , 417 , 666 , 667
<code>\@gls@between</code> .. 270 , 592 , 593 , 846	<code>\@gls@define@glossaryentrycounter</code> .. 14 , 14 , 42 , 226 , 227 , 377 , 393 , 500 , 501 , 626 , 627 , 643 , 752 , 753
<code>\@gls@body</code> .. 166 , 464 , 715 , 716	<code>\@gls@define@glossarysubentrycounter</code> .. 15 , 42 , 226 , 227 , 377 , 393 , 500 , 501 , 627 , 643 , 752 , 753
<code>\@gls@checkactual</code> .. 126 , 130 , 186 , 444 , 448 , 475 , 695 , 699 , 727	<code>\@gls@defsort</code> .. 16– 19 , 99 , 378 , 379 , 426 , 627–629 , 677
<code>\@gls@checkboxar</code> .. 126 , 129 , 186 , 444 , 447 , 476 , 695 , 698 , 727	<code>\@gls@defsortcount</code> .. 16– 19 , 73 , 378 , 379 , 410 , 627–629 , 660
<code>\@gls@checkedmkidx</code> .. 125–132 , 185–187 , 443– 449 , 474–476 , 694–701 , 726–728	<code>\@gls@deprecated@acr@warning</code> .. 261 , 774 , 776 , 778 , 780 , 782 , 784
<code>\@gls@checkesc</code> .. 126 , 131 , 186 , 695 , 728	<code>\@gls@do@acronymsdef</code> .. 20 , 20 , 42 , 74 , 379 , 380 , 393 , 410 , 629 , 630 , 643 , 660
<code>\@gls@checkescactual</code> .. 126 , 128 , 186 , 444–446 , 475 , 695–697 , 727	<code>\@gls@do@automake@err</code> .. 36 , 37 , 38
<code>\@gls@checkescbar</code> .. 126 , 128 , 186 , 444 , 446 , 476 , 695 , 697 , 727	<code>\@gls@do@glset</code> .. 119 , 119 , 135–140
<code>\@gls@checkesclevel</code> .. 126 , 129 , 186 , 446 , 447 , 695 , 697 , 698 , 728	<code>\@gls@do@indexdef</code> .. 40– 43 , 74 , 392 , 393 , 410 , 642 , 643 , 660
<code>\@gls@checkescquote</code> .. 126 , 127 , 186 , 187 , 444 , 445 , 475–477 , 695 , 696 , 727 , 728	<code>\@gls@do@numbersdef</code> .. 40 , 42 , 74 , 391–393 , 410 , 642 , 643 , 660
<code>\@gls@checklevel</code> .. 126 , 130 , 186 , 444 , 447 , 476 , 695 , 698 , 699 , 727	<code>\@gls@do@symbolsdef</code> .. 40 , 74 , 391 , 410 , 641 , 660
<code>\@gls@checkmkidxchars</code> .. 102 , 125 , 126 , 185 , 195 , 206 , 208 , 427 , 443 , 444 , 475 , 481 , 487 , 489 , 572 , 678 , 694 , 695 , 727 , 733 , 739 , 740 , 825	<code>\@gls@do@symbolssdef</code> .. 42 , 393 , 643
<code>\@gls@checkquote</code> .. 126 , 127 , 185 , 444 , 445 , 474 , 475 , 695 , 696 , 726 , 727	<code>\@gls@doautomake</code> .. 37 , 37 , 38 , 192 , 193 , 390 , 480 , 481 , 640 , 732 , 733
<code>\@gls@classI</code> .. 179 , 180 , 470 , 471 , 722	
<code>\@gls@classII</code> .. 180 , 471 , 722	
<code>\@gls@codepage</code> .. 62 , 215 , 404 , 492 , 654 , 744	

\@gls@docheckquotedef	\@gls@get@counterprefix	208, 488, 740
.....	185–187, 474–477, 726–729	\@gls@getbody 166, 464, 715
\@gls@docloadedfalse	... 6, 372, 622	\@gls@getcounter 74, 410, 660
\@gls@docloadedtrue	... 6, 372, 622	\@gls@getcounterprefix
\@gls@dodeflistparser	. 193, 481, 732	207, 207, 486–488, 738–740
\@gls@doentrycounterdef	\@gls@getgrouptitle	195, 233, 233,
.....	42, 43, 393, 643	270, 481, 504, 593, 733, 756, 846
\@gls@doentrydef	. 439, 440, 690, 691	\@gls@glossary	199, 199, 200, 485, 737
\@gls@dolast 209, 489, 741	\@gls@glslink@hyper@update@hook
\@gls@donext 209, 489, 741	118, 119
\@gls@donext@def	. 174, 468, 719, 720	\@gls@gobbleopt 72, 409, 659
\@gls@dosubentrycounterdef	\@gls@grptitle 233, 269,
.....	42, 43, 393, 643	270, 504, 592, 593, 756, 845, 846
\@gls@dothiswrite	\@gls@hyp@opt
.....	189–191, 478, 479, 730, 731	89, 90, 109–111, 120, 120, 121,
\@gls@elem 269, 592, 845	134–162, 241–243, 328–331,
\@gls@enablesavenonumberlist	419, 420, 432, 440, 450–463,
.....	79, 85, 412, 416, 663, 667	507, 508, 589–591, 670, 683,
\@gls@encapchar 128–	691, 702–714, 759, 760, 842–844
.....	130, 183, 207, 208, 446, 447,	\@gls@hyp@opt@cs 120, 440, 691
.....	473, 488, 489, 572, 575, 697,	\@gls@hypergroup	. 269, 270, 592, 845
.....	698, 724, 725, 740, 741, 825, 828	\@gls@ifinlist	. 55, 55, 399, 649, 650
\@gls@entry@count	\@gls@ifnotmeasuring	.. 103, 428, 679
.....	108, 109, 109, 431, 432, 682, 683	\@gls@igtype 75, 411, 661
\@gls@entry@field	88, 89, 107, 163,	\@gls@increment@currcount
.....	164, 165, 167–173, 338–340,	107, 109, 430, 431, 681, 682
.....	418, 419, 430, 463–467, 540,	\@gls@indexdef 40, 41, 392, 642
.....	541, 669, 681, 715–719, 793, 794	\@gls@initnonumberlist 78,
\@gls@escbsdq	79, 79, 96, 412, 423, 662, 663, 674
.....	125, 126, 184, 443, 444,	\@gls@islistofacronyms	21, 380, 630
.....	474, 576, 694, 695, 725, 726, 829	\@gls@keylist 369, 565, 818
\@gls@expand@fields	\@gls@keymap 79, 86, 86,
.....	82, 82, 83, 415, 665	87–89, 326, 336, 413, 416–418,
\@gls@expandonce 82, 415, 665	538, 587, 663, 667–669, 791, 840
\@gls@extramakeindexopts	\@gls@label 195,
.....	187, 192, 477, 480, 729, 732	198, 204, 206, 207, 481, 482,
\@gls@fetchfield	484, 487, 488, 733, 736, 738–740
.....	69, 87, 408, 417, 658, 668	\@gls@labelsfile 35, 389, 639
\@gls@field@link	\@gls@langmod 33,
.....	89, 90, 140, 141–155,	189, 190, 215, 478, 479, 730, 731
.....	353–358, 419, 420, 453–459,	\@gls@levelchar 103, 129,
.....	553–556, 670, 705–710, 806–809	130, 183, 428, 446, 447, 473,
\@gls@fieldaccess@display	575, 679, 697, 698, 724, 725, 828
.....	342, 343–345, 542–544, 795–797	\@gls@link	... 120, 122, 135–140,
\@gls@firsttok 221, 497, 749	156–163, 351–353, 441, 450–
\@gls@fixbraces	453, 459–463, 551, 552, 691,
.....	100, 209, 426, 489, 677, 741	692, 702–705, 710–715, 804–806
\@gls@forbidtexext	\@gls@link@checkfirsthyper
.....	71, 73, 409, 659, 660	121, 135–
\@gls@fulllist 35, 639	139, 441, 450–453, 692, 702–704

<code>\@gls@link@label</code>	<code>\@gls@noref@warn</code>
122, 441, 521, 526, 692, 773, 779	194, 198,
<code>\@gls@link@nocheckfirsthyper</code> ..	221, 481, 484, 497, 733, 736, 749
..... 140, 140, 156–	<code>\@gls@notranslatorhook</code>
163, 453, 459–463, 705, 710–714	29
<code>\@gls@link@opts</code>	<code>\@gls@numberlink</code> . 236, 237, 506, 758
122, 441, 521, 526, 692, 773, 779	<code>\@gls@numbersdef</code> 40, 40, 391, 392, 642
<code>\@gls@list</code> 269, 270, 592, 593, 845, 846	<code>\@gls@numlist@lastsep</code>
<code>\@gls@listsuffix</code>	174, 468, 719, 720
55, 399, 649	<code>\@gls@numlist@nextsep</code>
<code>\@gls@loadlist</code>	174, 468, 719, 720
.. 13, 13, 262, 376, 533, 626, 786	<code>\@gls@numlist@sep</code> 174, 468, 719, 720
<code>\@gls@loadlong</code>	<code>\@gls@old@chapter</code> 43, 393, 643
12,	<code>\@gls@oldnewglossaryentryposthook</code>
12, 13, 262, 376, 533, 625, 626, 786 337, 538, 792
<code>\@gls@loadsuper</code>	<code>\@gls@oldnewglossaryentryprehook</code>
12, 336, 337, 538, 791
12, 13, 262, 376, 533, 626, 786	<code>\@gls@onlypremakeg</code>
<code>\@gls@loadtree</code>	43, 43, 393, 643, 644
.. 13, 13, 262, 376, 533, 626, 786	<code>\@gls@order</code> 189–191, 478, 479, 729–731
<code>\@gls@local@increment@currcount</code>	<code>\@gls@org@LT@output</code> ... 288, 601, 855
.... 107, 109, 430, 431, 681, 682	<code>\@gls@org@glsnoidxdisplayloc</code> ..
<code>\@gls@loclist</code>	197, 483, 735
.... 196, 197, 222, 223, 482,	<code>\@gls@org@glsseeformat</code> 197, 483, 735
483, 497, 498, 734, 735, 749, 750	<code>\@gls@override@glossary</code>
<code>\@gls@map</code> .. 86, 87, 416, 417, 667, 668	39, 39, 40, 390, 391, 641
<code>\@gls@missinglang@warn</code>	<code>\@gls@patchtabularx</code> ... 103, 428, 679
... 23, 23, 47, 382, 395, 631, 645	<code>\@gls@preglossaryhook</code>
<code>\@gls@missingnumberlist</code>	212, 214, 490, 491, 742, 743
.... 81, 99, 414, 425, 664, 676	<code>\@gls@prevlevel</code> .. 299, 300, 321–
<code>\@gls@noaccess</code>	323, 582, 583, 608, 609, 620,
336, 342, 538, 541, 542, 791, 795	621, 835, 836, 861, 862, 874, 875
<code>\@gls@noexpand@field</code>	<code>\@gls@provide@newglossary</code>
81	71, 73, 409, 659, 660
<code>\@gls@noexpand@fields</code>	<code>\@gls@quotechar</code>
81, 83, 414, 415, 664, 665	127–131,
<code>\@gls@nohyperlist</code>	183–185, 187, 444–448, 473–
..... 22, 22, 74, 75, 122,	476, 575, 696–699, 724–728, 828
381, 410, 411, 441, 631, 661, 692	<code>\@gls@reference</code> 195, 198, 224, 481,
<code>\@gls@noidx@do</code> 221, 222, 497, 748, 749	482, 484, 499, 733, 734, 736, 751
<code>\@gls@noidx@getgrouptitle</code>	<code>\@gls@removedoption</code> .. 27, 31, 32, 38
.... 195, 233, 481, 504, 733, 756	<code>\@gls@removespaces</code> 237, 506, 758
<code>\@gls@noidx@sanitizesort</code>	<code>\@gls@renewglossary</code>
.... 25, 197, 383, 483, 632, 735 188, 200, 477, 485, 729, 737
<code>\@gls@noidx@setsanitizesort</code> 28,	<code>\@gls@replacementtext</code>
28, 197, 385, 483, 634, 635, 735 341, 342, 541, 794, 795
<code>\@gls@noidx@loclist@finalsep</code> ...	<code>\@gls@requested@languages</code> 31, 31, 45
..... 196, 483, 734, 735	<code>\@gls@rest</code>
<code>\@gls@noidx@loclist@prev</code> 196, 223,	166, 464, 715, 716
224, 483, 498, 499, 735, 750, 751	<code>\@gls@restore@glslocal</code> ... 119, 123
<code>\@gls@noidx@loclist@sep</code>	<code>\@gls@restoreat</code>
196, 223, 224,	85, 416, 667
482, 483, 498, 499, 734, 750, 751	<code>\@gls@roman</code> 58, 401, 402, 573, 652, 826
<code>\@gls@noist@toolate</code> ... 183, 184, 188	<code>\@gls@sanitized@tmp</code>
	125, 126, 443, 444, 694

\@gls@sanitizedesc	444, 485, 592, 593, 623, 628,
..... 24, 382, 387, 632, 637	645, 652, 653, 665, 694, 737, 846
\@gls@sanitizename 25, 382, 632	\@gls@tmpb
\@gls@sanitizesort	127, 128–132, 185, 187, 444–
..... 16, 25, 378, 382, 627, 632	449, 474–476, 695–701, 726–728
\@gls@sanitizesymbol	\@gls@toc 52, 53, 54, 398, 399, 648, 649
..... 25, 382, 387, 388, 632, 637	\@gls@type ... 191, 193, 194, 246,
\@gls@save@glsllocal ... 119, 119, 122	320, 480, 481, 510, 520, 522,
\@gls@saveentrycounter ... 123,	524, 525, 527, 530, 531, 533,
123, 175, 442, 468, 692, 693, 720	619, 732, 733, 762, 772, 775,
\@gls@savenonumberlist	776, 778, 780, 783, 784, 786, 873
... 78, 78, 79, 197, 412, 662, 663	\@gls@updatechecked
\@gls@see@noindex 125, 126, 127, 186, 443,
. 9, 9, 78, 374, 375, 412, 624, 662	444, 475, 476, 694, 695, 727, 728
\@gls@setacrstyle	\@gls@usetranslator
31,	29, 30,
42, 387, 388, 392, 393, 637, 643	31, 46, 386, 387, 394, 635, 636, 644
\@gls@setcounter ... 73, 74, 410, 660	\@gls@value
\@gls@setdefault@glslink@opts .	81, 82,
..... 122, 122, 441, 692	169, 414, 415, 465, 664, 665, 717
\@gls@setsort 16–19, 123, 175, 378,	\@gls@warnonglossdefined
379, 442, 468, 627–629, 692, 720	.. 24, 24, 211, 382, 490, 632, 742
\@gls@setupshortcuts 42, 392, 393, 643	\@gls@warnonthe glossdefined ...
\@gls@setupsort@clear 24, 24, 229, 382, 502, 632, 754
18	\@gls@write@entrycounts
\@gls@setupsort@def ... 17, 378, 628 108, 109, 431, 432, 682
\@gls@setupsort@none .. 18, 379, 629	\@gls@writedef 84, 85, 86, 416, 666, 667
\@gls@setupsort@standard	\@gls@writeisthook
..... 16, 377, 378, 627	182, 184, 184, 473, 474, 724, 726
\@gls@setupsort@use ... 17, 378, 628	\@gls@x
\@gls@sort	237, 758
\@gls@sort@A	\@gls@xdy@Lclass@Alpha-page-numbers
217, 218, 494, 746 59
\@gls@sort@B	\@gls@xdy@Lclass@Appendix-page-numbers
217, 218, 494, 746 59
\@gls@startswithexpandonce	\@gls@xdy@Lclass@Roman-page-numbers
..... 82, 82, 415, 665 58
\@gls@storenonumberlist	\@gls@xdy@Lclass@alpha-page-numbers
.. 79, 99, 412, 425, 662, 663, 676 59
\@gls@symbolsdef ... 40, 40, 391, 641	\@gls@xdy@Lclass@arabic-page-numbers
\@gls@this 59
201, 486, 738	\@gls@xdy@Lclass@arabic-section-numbers
\@gls@thisHloc 59
208, 488, 740	\@gls@xdy@Lclass@roman-page-numbers
\@gls@thisentryname 58
35, 639	\@gls@xdy@locationlist 58,
\@gls@thisfield 69, 70, 408, 658	179, 180, 401, 470, 471, 651, 722
\@gls@thislabel	\@gls@xdycheckbackslash
67, 209,	125, 132, 443, 449, 694, 700, 701
220, 406, 489, 496, 657, 741, 748	\@gls@xdycheckquote
\@gls@thislang 125, 131, 443, 448, 694, 700
45	\@gls@xxref
\@gls@thislist ... 174, 468, 719, 720	208, 489, 740, 741
\@gls@thisloc	
208, 488, 740	
\@gls@thisval	
87, 417, 668	
\@gls@title	
51, 397, 647	
\@gls@tmp	
8, 18, 46, 58,	
60, 82, 125, 126, 200, 237, 270,	
374, 379, 394, 402, 415, 443,	

<code>\@glsAlphacompositor</code>	49,	<code>\@glsminrange</code>	
	49 , 59 , 396 , 402 , 573 , 646 , 652 , 827	 178 , 178 , 179 , 180 , 470 ,
<code>\@glsHlocref</code>			471 , 573 , 574 , 721–723 , 826 , 827
 203 , 206 , 486 , 487 , 738 , 739	<code>\@glsname</code>	144 , 455 , 706
<code>\@glsacronymlists</code>	21 ,	<code>\@glsname@</code>	
	21 , 22 , 63 , 244 , 246 , 266 , 380 ,		144 , 145 , 355 , 455 , 553 , 706 , 807
	381 , 405 , 509 , 510 , 520 , 522–	<code>\@glsnavhypertarget</code> 269 , 269 , 592 , 845	
	525 , 527 , 529–531 , 533 , 535 ,	<code>\@glsnextpages</code>	
	630 , 655 , 761 , 762 , 772 , 774–	 213 , 227 , 491 , 501 , 743 , 752
	776 , 778 , 780 , 782–784 , 786 , 788	<code>\@glsnodesc</code>	80 , 95 , 97 , 99 ,
<code>\@glsaddkey</code>	88 , 89 , 418 , 669		413 , 423 , 426 , 664 , 673 , 674 , 676
<code>\@glsaddstoragekey</code>		<code>\@glsnoname</code>	80 , 95 , 97 , 99 ,
 87 , 88 , 418 , 668 , 669		413 , 423–425 , 663 , 673 , 675 , 676
<code>\@glsaddxdyattribute</code> . .	56 , 400 , 650	<code>\@glsnonextpages</code>	
<code>\@glsdefaultplural</code>	80 , 414 , 664	 213 , 227 , 491 , 500 , 743 , 752
<code>\@glsdefaultsort</code>		<code>\@glsnumberformat</code>	
 16 , 81 , 378 , 414 , 627 , 664	 118 , 122 , 175 , 198 , 207 ,
<code>\@glsdesc</code>	145 , 455 , 707		440 , 441 , 468 , 484 , 488 , 572 ,
<code>\@glsdesc@</code>			691 , 692 , 720 , 736 , 739 , 740 , 825
	145 , 146 , 355 , 455 , 554 , 707 , 807	<code>\@glsopenfile</code>	
<code>\@glsdescplural</code>	146 , 455 , 707		188 , 189 , 199 , 477 , 485 , 729 , 736
<code>\@glsdescplural@</code>		<code>\@glsorder</code>	
	146 , 147 , 355 , 455 , 554 , 707 , 807	 32 , 192 , 193 , 388 , 480 , 637 , 732
<code>\@glsdisp</code>	139 , 139 , 140 , 453 , 704	<code>\@glspl</code>	137 , 451 , 703
<code>\@glsentry</code>	35 , 105 , 106 , 109 ,	<code>\@glspl@</code>	108 , 111 , 137 ,
	389 , 429 , 430 , 432 , 639 , 680–683		137 , 329–331 , 431 , 432 , 451 ,
<code>\@glsentrytitlecase</code>			589–591 , 682 , 683 , 703 , 843 , 844
 169 , 169 , 465 , 466 , 717	<code>\@glsplural</code>	142 , 454 , 706
<code>\@glsfirst</code>	141 , 454 , 705	<code>\@glsplural@</code>	
<code>\@glsfirst@</code>			142 , 143 , 354 , 454 , 553 , 706 , 806
	141 , 142 , 354 , 454 , 553 , 705 , 806	<code>\@glsprefix@record@hook</code> . .	328 ,
<code>\@glsfirstletter</code>			328 , 329–332 , 589–591 , 842–845
 177 , 177 , 178 , 469 , 721	<code>\@glsreset</code>	
<code>\@glsfirstplural</code>	143 , 454 , 706		104 , 105 , 107 , 428–430 , 679–681
<code>\@glsfirstplural@</code>	143 ,	<code>\@glssee</code> .	100 , 209 , 426 , 489 , 677 , 741
	144 , 354 , 354 , 454 , 553 , 706 , 806	<code>\@glsshowaccsupp</code>	7 , 9 ,
<code>\@glshypernumber</code>			334 , 373 , 374 , 536 , 623 , 624 , 789
 236 , 236 , 506 , 757 , 758	<code>\@glsshowtarget</code>	7 ,
<code>\@glsisacronymlistfalse</code> 22 , 381 , 630			8 , 8 , 373 , 374 , 449 , 622 , 623 , 701
<code>\@glsisacronymlisttrue</code> 22 , 381 , 630		<code>\@glssymbol</code>	147 , 456 , 707
<code>\@glslink</code>	123 , 133 ,	<code>\@glssymbol@</code>	
	134 , 175 , 269 , 442 , 449 , 450 ,		147 , 148 , 356 , 456 , 554 , 707 , 807
	468 , 592 , 693 , 701 , 702 , 720 , 845	<code>\@glssymbolplural</code>	148 , 456 , 708
<code>\@glslocalreset</code>		<code>\@glssymbolplural@</code>	
	104 , 105 , 107 , 429 , 430 , 679–681		148 , 149 , 356 , 456 , 554 , 708 , 807
<code>\@glslocalunset</code>		<code>\@glstarget</code> . . .	133 , 134 , 230 , 269 ,
	104 , 105 , 107 , 429 , 430 , 680 , 681		450 , 502 , 592 , 701 , 702 , 754 , 845
<code>\@glslocref</code> 198 , 203 , 206 , 207 , 484 ,		<code>\@glstext</code>	140 , 453 , 705
486–488 , 572 , 736 , 738–740 , 825		<code>\@glstext@</code>	
			140 , 141 , 353 , 453 , 553 , 705 , 806

<code>\@glsunset</code>	405, 409, 418, 440, 469, 507,
104, 105 , 107, 429, 430, 680, 681	655, 659, 668, 669, 691, 721, 759
<code>\@glsuseri</code>	<code>\@ifundefined</code>
<code>\@glsuseri@</code> 45, 278, 289, 301, 308, 322,
149, 150, 356 , 456, 555, 708, 808	394, 583, 592, 596, 602, 609,
<code>\@glsuserii</code>	613, 620, 621, 644, 836, 845,
<code>\@glsuserii@</code>	850, 856, 862, 863, 866, 867, 874
150, 151, 357 , 457, 555, 708, 808	<code>\@ignored@glossaries</code> 35, 74,
<code>\@glsuseriii</code>	75, 75 , 389, 410, 411, 639, 660, 661
<code>\@glsuseriii@</code>	<code>\@input@</code>
151, 152, 357 , 457, 555, 709, 808	214, 492, 743
<code>\@glsuseriv</code>	<code>\@istfilename</code>
<code>\@glsuseriv@</code> 48, 192, 193, 396, 480, 646, 732
152, 153, 357 , 458, 555, 709, 808	<code>\@makecol</code>
<code>\@glsuserv</code>	288, 601, 602, 855
<code>\@glsuserv@</code>	<code>\@makeglossaries@warn@noprintglossary</code>
153, 154, 358 , 458, 555, 710, 809 193, 194 , 732, 733
<code>\@glsuservi</code>	<code>\@makeglossary</code> . 34, 188 , 193, 388,
<code>\@glsuservi@</code>	389, 477, 480, 638, 639, 729, 732
154, 155, 358 , 458, 556, 710, 809	<code>\@minus</code>
<code>\@glswidestname</code> 320, 320 , 321, 322,	274, 295,
582, 583, 620, 621, 836, 873, 874	314, 595, 605, 616, 848, 859, 870
<code>\@glswritefiles</code>	<code>\@mkboth</code>
.... 38, 198 , 390, 484, 640, 736	52, 397, 398, 647, 648
<code>\@glstr@doaccsupp</code> 333, 535, 789	<code>\@newglossary</code>
<code>\@glstr@record</code> 71, 73, 73 , 409, 410, 659, 660
7, 16–18, 34–36, 86,	<code>\@newglossaryentry@defcounters</code>
103, 125, 177, 181, 193, 195, 100,
274, 373, 378, 379, 388, 389,	101 , 107, 426, 427, 430, 677, 681
416, 417, 428, 443, 469, 471,	<code>\@newglossaryentryposthook</code>
472, 480, 481, 571, 574, 575, 88, 89,
622, 623, 627–629, 638, 639,	100, 101 , 326, 337, 418, 426,
667, 668, 679, 694, 720, 721,	427, 538, 588, 669, 677, 792, 841
723, 732, 733, 824, 827, 828, 848	<code>\@newglossaryentryprehook</code>
<code>\@gobblethree</code>	88, 89, 94, 96, 101 , 326, 336,
7, 373, 623	337, 418, 422, 423, 427, 538,
<code>\@idxitem</code>	587, 669, 673, 674, 677, 791, 841
315, 616, 870	<code>\@nil</code>
<code>\@ifclassloaded</code>	21, 100, 124–
6,	127, 166, 185, 186, 207, 209,
14, 51, 372, 376, 397, 622, 626, 647	221, 222, 236, 237, 381, 426,
<code>\@ifglossaryexists</code>	443, 444, 464, 475, 476, 488,
64,	489, 497, 498, 506, 572, 630,
64 , 211, 405, 490, 655, 656, 742	677, 694, 695, 715, 727, 728,
<code>\@ifl@t@r</code>	739, 741, 749, 750, 757, 758, 825
27, 384, 634	<code>\@nnil</code> 21, 209, 381, 489, 630, 741
<code>\@ifnextchar</code> 73, 120, 410, 440, 660, 691	<code>\@no@makeglossaries</code>
<code>\@ifpackageloaded</code> 6, 11, 12, 29–31,	. 34, 193, 195, 198 , 388, 389,
46, 62, 103, 173, 185, 332, 372,	481, 482, 484, 638, 732, 734, 736
375, 386, 387, 394, 404, 428,	<code>\@no@post@desc</code> 47 , 396, 577, 646, 830
467, 474, 535, 622, 625, 635–	<code>\@nopostdesc</code> 47 , 213, 395, 491, 646, 743
637, 644, 655, 679, 719, 726, 788	<code>\@onelevel@sanitize</code>
<code>\@ifstar</code> 8, 25, 35, 58, 86, 102, 125,
72, 87, 88, 120, 177, 178, 239,	177, 178, 182, 208, 210, 221,
	237, 374, 383, 402, 416, 428,

443, 444, 469, 470, 472, 489,	\@printgloss@checkexists@noignored
490, 497, 573, 574, 623, 632, 211, <u>211</u> , 212, 490, 742
633, 639, 652, 667, 678, 694,	\@printgloss@setsort
721, 724, 740, 742, 749, 826, 828 193, 195, <u>212</u> , 213, 481,
\@onlypreamble 73, 83,	482, 490, 491, 732, 734, 742, 743
94, 95, 109, 112, 194, 197, 410,	\@printglossary 211, 212, <u>212</u> , 490, 742
416, 422, 431, 433, 481, 483,	\@roman 58, 401, 573, 652, 826
660, 666, 673, 682, 684, 733, 735	\@secondofthree .. 120, 134, 136,
\@onlypremakeg <u>43</u> ,	138, 156, 158, 160, 162, 351,
48, 49, 55, 57, 60, 73, 184,	440, 450–452, 459–461, 463,
393, 396, 400, 403, 410, 474,	551, 691, 702, 704, 711–714, 804
643, 646, 650, 651, 653, 660, 726	\@secondoftwo
\@org@Gls@acrentryname <u>166</u>	26, 29–31, 46, 86, 87, 133–136,
\@org@glossaryentrynumbers	139, 156, 157, 159–161, 203,
..... 213, 214, 491, 743	274, 324, 351–353, 383, 384,
\@org@Gls@assign@descplural ...	386, 387, 394, 416, 417, 450,
..... 520, 527,	451, 453, 459, 461, 462, 551,
529, 531, 565, 568, 569, 772,	552, 570, 587, 633, 635–637,
779, 780, 782, 784, 819, 821, 822	644, 667, 668, 701–704, 710,
\@org@Gls@assign@firstpl	711, 713, 804–806, 823, 840, 848
..... 520–523, 525,	\@set@Glo@numformat <u>124</u> ,
527, 529–531, 565–569, 771,	207, 443, 488, 572, 694, 740, 825
772, 774–780, 782, 784, 818–822	\@sglsaddkey 88, 418, 669
\@org@Gls@assign@plural 520–523,	\@sglsaddstoragekey ... 87, 418, 668
525, 527, 529–531, 565–569,	\@tabacckludge 26, 383, 633
772, 774–780, 782, 784, 819–822	\@text@composite@x 26, 383, 633
\@org@Gls@assign@symbolplural .	\@thirdofthree ... 120, 136, 139,
..... 520–523, 525,	157, 159, 161, 163, 352, 440,
529–531, 566, 567, 569, 772,	451, 452, 459, 461–463, 551,
774–778, 782, 784, 819, 820, 822	691, 703, 704, 711–713, 715, 805
\@org@Gls@numberformat . 174, 467, 719	\@this@attr 181, 471, 472, 723
\@org@newglossaryentryprehook .	\@this@childlabel 216, 493, 745
..... 94, 422, 673	\@this@counter 56, 400, 650
\@outputpage 288, 601, 602, 855	\@this@ctr 181, 471, 723
\@p@glossarysection	\@this@key 87, 417, 668
..... 51, <u>53</u> , 397, 398, 647, 648	\@this@label .. 215, 216, 493, 744, 745
\@pgls 328, <u>328</u> , 589, 842	\@thiscs 43, 393, 394, 644
\@pgls@ 328, <u>328</u> , 589, 842	\@tmp 402, 573, 826
\@pglspl 329, <u>329</u> , 589, 842	\@use@option 41, 392, 642
\@pglspl@ 329, <u>329</u> , 589, 842	\@warn@nomakeglossaries
\@plus 274, 295, 191, 215, 480, 492, 731, 744
314, 595, 605, 616, 848, 859, 870	\@wrglossary@pageformat
\@print@glossary 202, <u>203</u> , 486, 738
.... 211, <u>214</u> , 490, 492, 742, 743	\@wrglossarynumberhook
\@print@noidx@glossary	202, <u>202</u> , 206, 486, 487, 738, 739
.... 212, <u>220</u> , 490, 496, 742, 748	\@write@Gls@labelsname .. 35, 36, 639
\@printgloss@checkexists .. 211,	\@xdy@main@language . 33, <u>33</u> , 388,
<u>211</u> , 212, 213, 490, 491, 742, 743	478, 479, 492, 638, 730, 731, 744
\@printgloss@checkexists@allowignored	\@xdy@attributelist <u>54</u> , 56,
..... <u>211</u> , 490, 742	181, 399, 400, 471, 649, 650, 723

\@xdyattributes	27, 384, 633
.. 54, 56, 179, 399, 400, 470,	258, 519, 771
571, 573, 649, 650, 722, 824, 826	258, 519, 771
\@xdycounters	access (key) 334
181, 399, 400, 471, 649, 650, 723	\accsuppglossaryentryfield
\@xdycrossrefhook	359, 536, 557, 789, 810
59, 180, 402, 471, 653, 723	\accsuppglossarysubentryfield
\@xdylanguage	359, 536, 557, 789, 810
62, 214, 215, 404, 492, 654, 744	\Acf
\@xdylettergroups	258, 519, 771
63, 63,	\acf
182, 405, 473, 575, 655, 724, 828	257, 519, 771
\@xdylocationclassorder	\Acfp
60, 61,	258, 519, 771
180, 403, 471, 574, 653, 723, 827	\acfp
\@xdylocoref	257, 519, 771
55, 56, 182, 399, 400, 472,	\Acl
571, 575, 649, 650, 724, 824, 828	257, 519, 771
\@xdynumbergrouporder	\acl
63,	257, 519, 771
178, 178, 405, 469, 470, 655, 721	\Aclp
\@xdyrequiredstyles	257, 519, 771
61, 61,	\aclp
179, 404, 470, 572, 654, 722, 826	257, 519, 771
\@xdysortrules	\Acp
61, 61,	258, 519, 771
182, 403, 473, 575, 654, 724, 828	\acp
\@xdystyle	258, 519, 771
179, 470, 572, 573, 722, 826	\acrfootnote
\@xdyuseralphabets	259, 520, 521, 526, 773, 779
57, 58,	\ACRfull
179, 401, 470, 573, 651, 722, 826	242, 508, 760
\@xdyuserlocationdefs	\Acrfull
59, 60, 180, 402, 403, 471,	242, 258, 508, 519, 759, 771
571, 574, 652, 653, 723, 825, 827	\acrfull
\@xdyuserlocationnames	241, 242, 257, 507, 519, 759, 771
59,	\ACRfullfmt
60, 402, 403, 572, 652, 653, 825	241, 242, 245, 253, 255, 364,
\@xfor@nextelement	366, 508, 510, 516, 517, 561,
209, 489, 741	563, 760, 761, 768, 769, 814, 816
\@xp@gl@getcounterprefix	\Acrfullfmt
203, 206, 207	241, 242, 245, 253, 255, 364,
\@xp@glshypernumber	366, 508, 509, 516, 517, 561,
236, 236	563, 760, 761, 768, 769, 814, 816
\\	\acrfullfmt
101, 125,	241, 242, 245, 253, 255, 364,
131, 177, 183, 184, 236, 237,	366, 507, 509, 516, 517, 561,
427, 443, 469, 473, 474, 506,	562, 759, 761, 767, 769, 814, 816
575, 576, 578, 579, 585, 586,	\acrfullformat
678, 694, 699, 700, 721, 725,	173, 240, 241, 467, 507,
758, 828, 829, 831, 832, 838, 839	508, 519, 532, 719, 759, 771, 785
\~	\ACRfullpl
26, 35, 383, 633, 639	243, 508, 760
\‘	\Acrfullpl
26, 383, 633	243, 258, 508, 519, 760, 771
\	\acrfullpl
126, 128, 129, 131, 186,	242, 243, 257, 508, 519, 760, 771
444, 446, 476, 695, 697, 699, 727	\ACRfullplfmt
\~	241, 243, 245, 253, 255, 365,
26, 384, 633	367, 509, 510, 516, 517, 561,
	563, 760–762, 768, 769, 814, 816
A	\Acrfullplfmt
\A	241, 243, 245, 253, 255, 365,
233	
\a	
26, 384, 633	
\AA	
27, 384, 633	

367, 508, 510, 516, 517, 561,
 563, 760, 762, 768, 769, 814, 816
 \acrfullplfmt
 241, 243, 245, 253, 255, 364,
 367, 508, 510, 516, 517, 561,
 563, 760, 762, 768, 769, 814, 816
 \acrlinkfootnote . 259, 259, 520, 773
 \acrlinkfullformat
 240, 241, 507–509, 759–761
 \ACRlong 160, 462, 713
 \Acrlong
 160, 160, 257, 461, 519, 713, 771
 \acrlong . 159, 257, 461, 519, 712, 771
 \ACRlongpl 162, 463, 714
 \Acrlongpl
 162, 162, 257, 462, 519, 714, 771
 \acrlongpl 161, 257, 462, 519, 714, 771
 \acnameformat
 244, 509, 524, 567, 761, 777, 820
 \acnolinkfootnote 259, 520, 773
 acronym (option) 20
 \acronymentry 244,
 245, 247–252, 254–256, 360–
 363, 365–369, 509–519, 557–
 564, 761–771, 811–813, 815–818
 \acronymfont
 . 115, 116, 156–159, 166, 173,
 243, 243, 244, 245, 247–256,
 348, 349, 351–353, 360–369,
 437, 438, 459–461, 464, 467,
 509–519, 521, 522, 524–530,
 549–552, 557–564, 566–568,
 688, 689, 710–712, 716, 719,
 761–771, 773, 775, 776, 778–
 781, 783, 802–806, 811–819, 821
 acronymlists (option) 22
 \acronymname 20, 44, 47, 379,
 380, 394, 395, 629, 630, 644, 645
 acronyms (option) 20
 \acronymsort 244, 245, 247–252, 254,
 256, 360–363, 365, 366, 368,
 369, 509–519, 557–559, 562,
 564, 761–771, 811–813, 815–818
 \acronymtype 19,
 20, 244, 246, 379, 380, 509,
 510, 519–533, 565–568, 629,
 630, 761, 762, 771–786, 818–821
 \acrpluralsuffix
 . 240, 245, 247–250, 254–256,
 360, 361, 365–368, 507, 509,
 511–513, 516–523, 525–527,
 529, 530, 532, 557, 558, 562,
 564–569, 759, 761, 763–765,
 768–771, 773–775, 777–780,
 782, 783, 785, 786, 811, 815–822
 \ACRshort 157, 459, 711
 \Acrshort
 156, 156, 257, 459, 519, 711, 771
 \acrshort 155, 257, 459, 519, 710, 771
 \ACRshortpl 158, 460, 712
 \Acrshortpl
 158, 158, 257, 460, 519, 712, 771
 \acrshortpl 157, 257, 460, 519, 711, 771
 acrstyles:
 dua 252, 363
 dua-desc 254, 365
 footnote 254, 366
 footnote-desc 256, 368
 footnote-sc 255, 367
 footnote-sc-desc 256, 368
 footnote-sm 255, 368
 footnote-sm-desc 256, 368
 long-sc-short 249
 long-sc-short-desc 250, 361
 long-short 247, 360
 long-short-desc 250, 361
 long-sm-short 249
 long-sm-short-desc 251, 362
 long-sp-short 247
 long-sp-short-desc 250
 sc-short-long 249
 sc-short-long-desc 251, 362
 short-long 248, 360
 short-long-desc 251, 362
 sm-short-long 249
 sm-short-long-desc 251, 363
 \Acs 257, 519, 771
 \acs 257, 519, 771
 \Acsp 257, 519, 771
 \acsp 257, 519, 771
 \add@accent@ 26, 383, 633
 \addcontentsline 54, 399, 649
 \addglossarytocaptions
 45, 46, 394, 644, 645
 \addtolength .. 322, 583, 621, 836, 874
 \advance
 17, 18, 97, 124, 288, 378, 379,
 424, 442, 601, 628, 675, 693, 855
 \AE 26, 384, 633
 \ae 26, 384, 633

altlist (style)	276	\begin	181, 221, 275,
altlistgroup (style)	276	278–284, 286, 287, 289–314,	
altlisthypergroup (style)	276	472, 496, 574, 595–599, 601–	
altlong4col (style)	283	616, 723, 748, 827, 848, 850–870	
altlong4col-booktabs (style)	286	\BeginAccSupp	333, 536, 789
altlong4colborder (style)	283	\begingroup	6, 8, 35, 200,
altlong4colheader (style)	283	205, 237, 372, 374, 485, 487,	
altlong4colheaderborder (style) .	284	506, 622, 623, 639, 737, 739, 758	
altlongragged4col (style)	292	\bfseries	
altlongragged4col-booktabs (style)	287	279, 281–283, 285, 286, 290–	
altlongragged4colborder (style) .	294	292, 294, 302, 304–307, 309–	
altlongragged4colheader (style) .	293	314, 597–600, 603–605, 610–	
altlongragged4colheaderborder (style)	294	616, 850–854, 856–859, 863–870	
\altnewglossary	73 , 410 , 660	\bgroup .	26, 94, 131, 174, 207, 213,
altsuper4col (style)	306	216, 237, 383, 422, 467, 491,	
altsuper4colborder (style)	307	493, 633, 673, 699, 719, 743, 745	
altsuper4colheader (style)	306	\boolean	532, 785
altsuper4colheaderborder (style)	307	\boolfalse	38, 390, 640
altsuperragged4col (style)	312	\booltrue	390, 640
altsuperragged4colborder (style)	313	\bottomrule ...	285, 286, 600, 853, 854
altsuperragged4colheader (style)	313	\box	288, 601, 602, 855
altsuperragged4colheaderborder (style)	313		
almtree (style)	320	C	
almtreegroup (style)	323	\c	26, 384, 633
almtreehypergroup (style)	323	\c@equation	124, 442, 693
\andname ..	45 , 210 , 394 , 489 , 644 , 741	\c@glossaryentry	14, 377, 626
\AnyTrackedLanguages ...	46 , 324 , 325 , 395 , 570 , 587 , 645 , 823 , 840	\c@glossarysubentry ...	15, 377, 627
\appto	22 , 27 , 79 , 88 , 89 , 203 , 326 , 336 , 381 , 384 , 413 , 418 , 538 , 587 , 588 , 631 , 634 , 663 , 669 , 791 , 840 , 841	\c@page	201–
\AtBeginDocument	20 , 62 , 85 , 103 , 176 , 195 , 380 , 404 , 416 , 428 , 468 , 482 , 629 , 655 , 667 , 679 , 720 , 734	203, 205, 206, 486, 487, 738, 739	
\AtEndDocument	35 , 37 , 85 , 108 , 194 , 198 , 214 , 215 , 270 , 389 , 390 , 416 , 431 , 481 , 484 , 492 , 592 , 639 , 640 , 667 , 682 , 733 , 736 , 744 , 845	\capitalisewords	169, 717
automake (option)	36	\catcode 35 , 85 , 131 , 416 , 639 , 667 , 699	
automakegloss (option)	37	\cGls	110 , 110 , 432 , 683
automakeglosslite (option)	37	\cgls	109 , 110 , 432 , 683
B		\cglsformat 108 , 110 , 431 , 432 , 682 , 683	
\b	26 , 384 , 633	\cglsformat 108 , 110 , 431 , 432 , 682 , 683	
		\cGlSpl	111 , 111 , 432 , 433 , 683
		\cGlSpl	110 , 111 , 432 , 683
		\cGlSplformat	
		108 , 111 , 431 , 433 , 682 , 684
		\cGlSplformat	
		108 , 111 , 431 , 432 , 682 , 683
		\char	233, 504, 756
		\cleardoublepage ..	53 , 398 , 648 , 649
		\clearpage	53 , 398 , 648 , 649
		\closeout ..	35 , 85 , 182 , 184 , 189 ,
		199, 389 , 416 , 473 , 474 , 477 ,	
		485, 639 , 667 , 724 , 726 , 729 , 737	
		\compattglossarystyle	
		576–586, 830–840
		compatible-2.07 (option)	38

compatible-3.07 (option)	38		
\compatibleglossentry	230 , 333 , 502 , 503 , 536 , 754 , 755 , 789		719 , 720 , 722 , 723 , 727–729 , 733 , 736–738 , 740–745 , 751 , 754–757 , 759 , 772 , 786–789 , 792–794 , 797 , 804–806 , 822– 825 , 836 , 841 , 845 , 846 , 873 , 874
\compatiblesubglossentry	231 , 333 , 503 , 536 , 755 , 789		266 , 535 , 788
\copy	288 , 601 , 602 , 855		
\count@	222 , 497 , 749		
counter (key)	77		45 , 47 , 50 , 71 , 78 , 81 , 82 , 89 , 90 , 163 , 189–191 , 216 , 219 , 220 , 222 , 224 , 226 , 247 , 326 , 395 , 397 , 409 , 414 , 415 , 419 , 420 , 439 , 440 , 478 , 479 , 493 , 495–497 , 499 , 500 , 505 , 511 , 577–586 , 588 , 645 , 647 , 659 , 664 , 665 , 670 , 690 , 730 , 731 , 745 , 747–749 , 751 , 752 , 757 , 763 , 831–835 , 837–841
counter (option)	22		
counters:			
glossaryentry	14		
glossarysubentry	15		
counterwithin (option)	14		
\csdef	24 , 88–90 , 100 , 101 , 106 , 107 , 215 , 216 , 235 , 246 , 382 , 418–420 , 426 , 427 , 430 , 431 , 493 , 505 , 510 , 511 , 576 , 632 , 669 , 670 , 677 , 678 , 681 , 744 , 745 , 757 , 762 , 830		99 , 109 , 425 , 431 , 676 , 682 , 772
\csedef	109 , 202 , 431 , 486 , 682 , 738		
\csgdef	19 , 51 , 71 , 74 , 107 , 108 , 210 , 224 , 397 , 409 , 410 , 430 , 431 , 490 , 499 , 647 , 659 , 660 , 681 , 682 , 742 , 751		
\cslet	78 , 79 , 94 , 101 , 220 , 413 , 422 , 427 , 496 , 663 , 673 , 678 , 748		
\csname	15–18 , 26 , 41 , 44 , 46 , 47 , 52 , 53 , 56 , 58 , 59 , 62 , 64 , 67 , 72–74 , 81 , 82 , 88–90 , 92 , 93 , 97 , 98 , 100–103 , 105 , 122–124 , 135–139 , 156–163 , 174 , 175 , 179–181 , 186–189 , 195 , 198– 200 , 202 , 207 , 208 , 212–214 , 217 , 234 , 235 , 239 , 262–267 , 269 , 320 , 322 , 326 , 337 , 338 , 341 , 345 , 351–353 , 370 , 371 , 377–379 , 383 , 392 , 394 , 395 , 398 , 400 , 402 , 404–406 , 409 , 410 , 414 , 418–421 , 424–429 , 439 , 441–443 , 450–453 , 459– 463 , 468 , 470–472 , 475–477 , 482 , 484–486 , 488 , 489 , 491– 493 , 499 , 502 , 503 , 505 , 507 , 533–536 , 538–541 , 544 , 551 , 552 , 569–572 , 583 , 588 , 592 , 593 , 620 , 621 , 627–629 , 633 , 642 , 644 , 645 , 648 , 650 , 652 , 654 , 655 , 657 , 659 , 660 , 664 , 665 , 668–672 , 675–680 , 690 , 692–694 , 702–704 , 710–715 ,		
			D
\d	26 , 383 , 633		
\day	179 , 183 , 470 , 473 , 572 , 575 , 722 , 725 , 826 , 828		
debug (option)	7		
\DeclareAcronymList	20 , 21 , 22 , 244 , 246 , 379–381 , 509 , 510 , 520 , 522 , 523 , 525 , 527 , 529 , 531 , 533 , 629–631 , 761 , 762 , 772 , 774 , 776 , 778 , 780 , 782 , 784 , 786		
\DeclareCurrentRelease	4 , 268 , 271 , 273 , 277 , 284 , 289 , 295 , 300 , 308 , 314 , 324 , 325 , 332 , 371 , 372 , 622		
\DeclareListParser	193 , 481 , 732		

<code>\DeclareOption</code>	11, 325, 332, 375, 535, 587, 625, 788, 840	<code>\define@boolkey</code>	10, 12–15, 20, 27, 28, 31–33, 38, 119, 226, 375–377, 379, 384, 385, 387, 388, 390, 440, 500, 624–627, 629, 634, 637, 638, 640, 691, 752
<code>\DeclareOptionX</code>	11, 375, 625	<code>\define@choicekey</code>	7, 9, 10, 15, 28, 30, 32, 36, 39, 41, 78, 225, 226, 373–375, 377, 385, 386, 388, 389, 391, 412, 499, 500, 622, 624, 625, 627, 635–637, 640, 641, 662, 751, 752
<code>\DeclareRelease</code>	4, 268, 271, 273, 277, 284, 289, 295, 300, 308, 314, 324, 325, 332, 371, 372, 622	<code>\define@key</code>	11, 14, 22, 23, 29, 31, 33, 75–80, 88, 89, 118, 175, 225, 226, 325, 334–336, 375, 377, 381, 385, 388, 411–413, 418, 440, 468, 499, 500, 536, 537, 587, 625, 627, 631, 635, 638, 661–663, 669, 691, 720, 751, 752, 790, 791, 840
<code>\DeclareRobustCommand</code>	47, 395, 528, 646, 780	<code>\DefineAcronymSynonyms</code>	42, 257, 258, 393, 519, 643, 771
<code>\def</code>	11, 12, 14–21, 25–27, 32, 33, 35, 37–40, 42–44, 47, 48, 51, 55, 57–63, 67, 71–80, 82, 92, 94–99, 101, 103, 108, 110–112, 118, 122–132, 135–139, 141–163, 174, 175, 178, 183, 185–187, 189–192, 194–196, 199, 202, 203, 205–209, 212, 215, 220–227, 233–238, 242–244, 270–273, 299, 300, 320–323, 325, 326, 328–332, 334–337, 351–358, 375–380, 383, 384, 388, 390–397, 399, 401–407, 409–413, 415, 421–428, 431–433, 439–463, 468, 470, 473–480, 482–484, 486–489, 491, 492, 496–500, 504–509, 520–525, 527–529, 531, 533, 536–538, 551–556, 565–569, 572, 575, 577, 582, 583, 587, 589–594, 608, 609, 620, 621, 625–630, 632–634, 638–641, 643–647, 649, 651–657, 659–663, 665, 671, 673–679, 681–684, 690–715, 719, 720, 722, 724, 726–731, 734–736, 738–744, 748–752, 756–761, 772, 774–782, 784, 786, 789–792, 804–809, 819–822, 825, 826, 828, 830, 836, 840–844, 846, 847, 861, 862, 873–875	<code>\delimN</code>	50, 182, 193, 224, 236, 397, 472, 481, 498, 506, 574, 647, 724, 732, 750, 758, 828
<code>\def@glx@xdycheckbackslash</code>	132, 133, 448, 449, 700, 701	<code>\delimR</code>	50, 182, 236, 397, 472, 506, 574, 647, 724, 758, 828
<code>\DefaultNewAcronymDef</code>	258, 369, 519, 520, 565, 771, 773, 818	<code>description (key)</code>	75
<code>\defglstdisplay</code>	118, 439, 690	<code>description (option)</code>	32
<code>\defglstdisplayfirst</code>	118, 439, 690	<code>descriptionaccess (key)</code>	335
<code>\defglstryfmt</code>	71, 73, 74, 246, 258, 409, 410, 439, 510, 519, 520, 522, 524, 526, 528, 530, 532, 659–661, 690, 762, 771, 773, 775, 776, 778, 781, 783, 785	<code>\DescriptionDUANewAcronymDef</code>	259, 522, 524, 775, 776
		<code>\DescriptionFootnoteNewAcronymDef</code>	259, 369, 521, 522, 566, 773, 774, 819
		<code>\descriptionname</code>	44, 47, 279, 281–283, 285, 286, 290–292, 294, 302, 304–307, 309–314, 394, 395, 597–600, 603–605, 610–612, 614–616, 644, 645, 850–854, 856–859, 863–870
		<code>\DescriptionNewAcronymDef</code>	260, 369, 524, 525, 566, 777, 778, 820
		<code>descriptionplural (key)</code>	75
		<code>descriptionpluralaccess (key)</code>	335
		<code>\DH</code>	27, 384, 634
		<code>\dh</code>	27, 384, 634
		<code>\dimen@</code>	248, 288, 320, 512, 601, 619, 620, 764, 855, 873
		<code>\directlua</code>	42
		<code>\disable@keys</code>	42, 392, 642

disablemakegloss (option)	34	\edef	43, 46, 54–56, 58, 60,
\do	31 , 41 , 43 , 45 , 55 , 56 , 63 , 64, 86, 87, 125, 174, 179–181, 191, 193, 201, 209, 215, 216, 246, 269, 270, 387, 392, 393, 399, 400, 405, 416, 417, 443, 468, 470, 471, 480, 481, 486, 489, 493, 510, 520, 522, 524, 525, 527, 530, 531, 533, 572, 592, 593, 637, 642, 644, 650, 655, 667, 668, 694, 719, 722, 723, 732, 733, 738, 741, 744, 745, 762, 772, 775, 776, 778, 780, 783, 784, 786, 826, 845, 846	\edef	43, 46, 54–56, 58, 60, 61, 72, 81, 82, 85, 87, 123–132, 177, 183–187, 189–191, 193, 199, 214, 215, 217, 218, 222, 233, 237, 239, 379, 380, 393, 395, 399–406, 409–411, 414– 417, 420–423, 427, 439, 441– 449, 467–469, 473–476, 478, 479, 481, 484, 487, 488, 490, 492–494, 497, 501, 504, 506, 507, 519, 521, 522, 524, 526, 528, 530, 565–568, 571–573, 575, 592, 644, 645, 649–654, 659, 660, 664, 665, 667, 668, 690, 693–701, 720, 721, 724, 726–728, 730–732, 736, 739, 740, 744, 746, 749, 756, 758, 759, 771, 773, 775, 777, 779, 781, 783, 818–821, 824–826, 828
\do@glo@storeentry	16–19, 100, 377–379, 426, 627–629, 677	\egroup	26, 94, 174, 207, 214, 216, 237, 383, 422, 468, 491, 493, 633, 673, 720, 743, 745
\do@gls@link@checkfirsthyper 120 , 122 , 135 – 140 , 156–163, 351–353, 441, 450– 453, 459–463, 551, 552, 691, 692, 702–705, 710–714, 804, 805	\else	8, 13– 15, 18–21, 23, 25, 27, 28, 35, 37, 38, 41–43, 48, 49, 52–58, 60–63, 71, 78, 81, 97, 98, 101– 103, 108, 119, 121–132, 166, 177–179, 182–185, 187–189, 191, 192, 199–203, 205–209, 213, 222, 226, 228, 229, 234, 236, 237, 248, 260, 262, 269, 275, 279, 280, 282, 285, 286, 288, 290, 291, 293, 302, 303, 305, 309, 311, 313, 316, 318, 319, 321, 322, 342, 373, 374, 376, 377, 379–382, 384, 385, 389–393, 396–405, 412, 414, 424, 425, 427, 428, 431, 441– 453, 464, 469, 470, 472–479, 484–489, 491, 497, 500–502, 504, 506, 512, 522, 523, 525, 527, 528, 530–533, 541, 572– 577, 581–583, 592, 595, 597, 598, 600–604, 609–611, 613– 615, 617–621, 623, 626–632, 634, 635, 639–644, 646–655, 662, 664, 675, 676, 678, 679, 682, 692–704, 715, 716, 721, 722, 724, 726–731, 736–741, 743, 749, 752–754, 756, 758,
\do@gls@xdycheckbackslash 125 , 443 , 694		
\do@gls@disablehyperinlist 122 , 122 , 441 , 692		
\do@gls@haschildren 67 , 406 , 407 , 656 , 657		
\do@write@gls@labelsname 35 , 35 , 36 , 639		
\doifglossarynoexistsordo 66 , 72 , 406 , 409 , 656 , 659		
\dtl@ifsingle	504, 756		
\dtl@insertinto	217 , 494 , 746		
\dtl@sortresult	217 , 218 , 494 , 746		
\dtlcompare	218 , 494 , 746		
\dtlicompare	218 , 494 , 746		
\DTLifinlist 75 , 122 , 411 , 441 , 661 , 692			
\DTLifint	233 , 504 , 756		
\dtlletterindexcompare 217 , 494 , 746			
\DTLsubstituteall	126 , 444 , 694		
\dtlwordindexcompare	217 , 494 , 746		
dua (acrstyle)	252 , 363		
dua (option)	32		
dua-desc (acrstyle)	254 , 365		
\DUANewAcronymDef 261 , 530 , 531 , 783 , 784		
E			
\eappto	410 , 411 , 427 , 486		

764, 774–776, 778, 780–783,	506, 622, 624, 639, 737, 739, 758
785, 786, 795, 825, 826, 828–	<code>\endhead</code> 279–283,
830, 834–836, 845, 848, 850–	285, 286, 290–292, 294, 597–
858, 863–865, 867–869, 871–875	600, 603–605, 850–854, 856–859
<code>\emph</code> 209, 238, 489, 507, 741, 759	<code>\endtheglossary</code> 6, 372, 622
<code>\empty</code> 237, 333, 506, 535, 758, 789	<code>entrycounter</code> (option) 14
<code>\encodingdefault</code> 26, 383, 633	<code>\entryname</code> 44, 47,
<code>\end</code> 181, 221, 275,	279, 281–283, 285, 286, 290–
278–284, 286, 287, 289–314,	292, 294, 302, 304–307, 309–
472, 497, 574, 595–599, 601–	314, 394, 395, 597–600, 603–
616, 723, 748, 827, 848, 850–870	605, 610–612, 614–616, 644,
<code>\end@doifinlist</code> 55, 399, 649	645, 850–854, 856–859, 863–870
<code>\end@getprefix</code> 208, 488, 740	environments:
<code>\end@glis@islistofacronyms</code>	<code>theglossary</code> 229
. 21, 380, 381, 630	<code>\equal</code> 29, 43, 53, 123, 193, 269, 385,
<code>\EndAccSupp</code> 333, 536, 789	393, 398, 442, 480, 504, 592,
<code>\endcsname</code> 15–18, 26, 41, 44, 46, 47,	635, 643, 648, 693, 732, 756, 845
52, 53, 56, 58, 59, 62, 64, 67,	<code>esclocations</code> (option) 12
72–74, 81, 82, 88–90, 92, 93,	<code>\expandafter</code> . 8, 16–18, 25, 26, 41,
97, 98, 100–103, 105, 122–124,	43, 45, 46, 56, 58, 59, 61–64,
135–139, 156–163, 174, 175,	67, 72–75, 81, 82, 86–90, 92,
179–181, 186–189, 195, 198–	93, 97, 98, 100, 102, 103, 105,
200, 202, 207, 208, 212–214,	117, 122–132, 164–166, 169,
217, 234, 235, 239, 262–267,	174, 177, 181, 185–188, 199–
269, 320, 322, 326, 337, 338,	201, 203, 205–207, 209, 213,
341, 345, 351–353, 370, 371,	216, 217, 221, 222, 230, 235,
377–379, 383, 392, 394, 395,	237, 239, 262–267, 269, 270,
398, 400, 402, 404–406, 409,	320, 326, 337, 338, 341, 342,
410, 414, 418–421, 424–429,	369–371, 374, 378, 379, 383,
439, 441–443, 450–453, 459–	392–395, 400, 402–406, 409–
463, 468, 470–472, 475–477,	411, 414–421, 424–429, 439,
482, 484–486, 488, 489, 491–	441–449, 464, 468, 469, 471,
493, 499, 502, 503, 505, 507,	472, 474–477, 484–489, 491,
533–536, 538–541, 544, 551,	493, 497, 498, 502, 503, 505–
552, 569–572, 583, 588, 592,	507, 521, 526, 533–536, 538–
593, 620, 621, 627–629, 633,	542, 565, 569–572, 574, 575,
642, 644, 645, 648, 650, 652,	588, 592, 593, 620, 623, 627–
654, 655, 657, 659, 660, 664,	629, 632, 633, 642, 644, 645,
665, 668–672, 675–680, 690,	650, 652, 654, 655, 657, 659–
692–694, 702–704, 710–715,	661, 664, 665, 667–672, 675–
719, 720, 722, 723, 727–729,	680, 690, 692–700, 715–717,
733, 736–738, 740–745, 751,	720, 721, 723, 726–729, 736–
754–757, 759, 772, 786–789,	741, 743, 745, 749, 750, 754,
792–794, 797, 804–806, 822–	755, 757–759, 772, 773, 779,
825, 836, 841, 845, 846, 873, 874	786–789, 792–795, 818, 822–
<code>\endfoot</code> 279–283,	825, 827, 828, 841, 845, 846, 873
285, 286, 290–292, 294, 597–	<code>\expandonce</code> 81, 82, 93, 125, 126, 186,
600, 603–605, 850–854, 856–859	187, 202, 217, 218, 244, 414,
<code>\endgroup</code> 6, 8, 35, 201,	415, 443, 444, 475–477, 486,
206, 237, 372, 374, 485, 487,	494, 502, 503, 509, 519, 521,

523, 525, 526, 529, 530, 536,
664, 665, 672, 694, 695, 727,
728, 738, 746, 754, 755, 761,
771–775, 777, 779, 782, 783, 789
\ExplSyntaxOff 5, 233, 234
\ExplSyntaxOn 4, 233

F

\fi 6–8, 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, 101–103, 107, 108,
119, 121–133, 166, 176–179,
182, 184, 185, 187, 188, 190–
194, 199–203, 205–210, 212,
213, 215, 222, 226, 228, 229,
234, 236, 237, 248, 258, 260,
262, 270, 275, 279, 280, 282,
285, 286, 288, 290, 292, 293,
302, 303, 305, 309, 311, 313,
316–322, 333, 342, 372–385,
387, 389–393, 395–405, 410,
412, 414, 424–428, 431, 441–
453, 464, 468–470, 472, 474–
478, 480, 481, 485–492, 497,
500–502, 504–507, 512, 519,
520, 522, 523, 525–533, 535,
536, 542, 571–577, 581–583,
592, 595, 597, 598, 600–604,
609–611, 613–615, 617–632,
634–636, 639–644, 646–655,
660, 662, 664, 672, 674–679,
682, 692–705, 716, 720–722,
724, 726–733, 736–744, 749,
752–754, 756–758, 764, 771,
772, 774–776, 778, 780–786,
788, 789, 795, 824–826, 828–
830, 834–836, 845, 848, 850–
858, 863–865, 867–869, 871–875
\findrootlanguage 62, 404, 654
first (key) 76
firstaccess (key) 334
\firstacronymfont
. 117, 243, 247–249, 254, 350,
351, 360, 361, 366, 439, 509,
511, 512, 517, 521, 524, 526,
528, 551, 557, 558, 562, 689,
690, 761, 763, 764, 769, 773,
777, 779, 781, 804, 810, 811, 815

firstplural (key) 76
firstpluralaccess (key) 334
\fmtversion 27, 384, 634
\footnote 254, 259,
366, 517, 520, 562, 769, 773, 815
footnote (acrstyle) 254, 366
footnote (option) 31
footnote-desc (acrstyle) . . . 256, 368
footnote-sc (acrstyle) 255, 367
footnote-sc-desc (acrstyle) . 256, 368
footnote-sm (acrstyle) 255, 368
footnote-sm-desc (acrstyle) . 256, 368
\FootnoteNewAcronymDef . . . 260,
369, 526, 527, 567, 779, 780, 821
\footnotesize 8, 374, 623
\forallacronyms 63, 405, 655
\forallglossaries
. . 63, 64, 198, 212, 320, 405,
484, 490, 619, 655, 736, 742, 873
\forallglsentries 35, 64, 105, 106,
109, 176, 389, 405, 429, 430,
432, 469, 639, 655, 680–682, 720
\ForEachTrackedDialect . 46, 324,
325, 395, 570, 587, 645, 823, 840
\forglsentries 63, 64, 67,
101, 220, 320, 405, 406, 427,
496, 619, 655, 657, 678, 748, 873
\forlistcsloop
. . . . 215, 221, 493, 497, 744, 748
\forlistloop
196, 197, 223, 483, 498, 735, 750

G

\gdef 17,
35, 56, 72, 92, 97, 98, 200, 227,
269, 378, 400, 409, 420, 424,
485, 500, 501, 592, 628, 639,
650, 659, 671, 675, 737, 752, 845
\Genacrformat
. . . . 116, 117, 245, 247, 248,
254, 350, 351, 360, 361, 366,
438, 439, 510–512, 517, 550,
551, 557, 558, 562, 689, 761–
764, 769, 803, 804, 810, 811, 815
\genacrformat
. . . . 116, 117, 117, 245, 247,
248, 254, 350, 350, 360, 366,
438, 439, 509–512, 517, 550,
557, 558, 562, 689, 690, 761–
764, 769, 803, 804, 810, 811, 815

<code>\GenericAcronymFields</code>	478, 479, 481–485, 488, 490,
. 245, <u>245</u> , 247, 248, 250–	491, 503, 505, 535, 571, 622–
254, 256, 360–366, 368, 369,	624, 631, 632, 634, 635, 638–
509–519, 557–559, 561, 562,	641, 647, 649, 656, 659, 662,
564, 761–770, 810–815, 817, 818	664, 666, 667, 683, 690, 691,
<code>\Genplacrfullformat</code>	719, 730, 731, 733–737, 740,
. 116, <u>117</u> , 245, 247–249, 254,	742, 743, 755, 757, 784, 789, 824
350, <u>351</u> , 360, 361, 366, 438,	<code>\GlossariesWarningNoLine</code>
439, 510–512, 517, 550, 551, 7, 23, <u>23</u> , 194, 195,
557, 558, 562, 689, 690, 762–	199, 215, 269, 373, 381, 382,
764, 769, 803, 804, 810, 811, 815	480–482, 484, 492, 592, 622,
<code>\genplacrfullformat</code> 116,	623, 631, 733, 734, 736, 744, 845
117, <u>117</u> , 245, 247, 248, 254,	<code>\glossary</code> 38,
350, <u>351</u> , 360, 361, 366, 438,	39, <u>39</u> , 390, 391, 572, 640, 641, 825
439, 510–512, 517, 550, 551,	<code>\glossaryentry</code> 207,
557, 558, 562, 689, 690, 762–	208, 488, 489, 572, 740, 741, 825
764, 769, 803, 804, 810, 811, 815	<code>glossaryentry</code> (counter) <u>14</u>
<code>\GetTitleStringDisableCommands</code>	<code>\glossaryentryfield</code> <u>231</u> , 359, 502,
. 274, 848	503, 557, 577–582, 584–586,
<code>\GetTitleStringSetup</code> 274, 848	754, 755, 810, 830–835, 837–839
<code>\glo@desc</code> 577, 830	<code>\glossaryentrynumbers</code>
<code>\glo@do@compare</code> . . 217, 218, 494, 746	. . <u>11</u> , 12, 181, 213, 214, 222,
<code>\glo@grabfirst</code>	223, 226, 227, 375, 376, 472,
. <u>221</u> , 222, 497, 498, 749, 750	491, 498, 500, 501, 574, 625,
<code>\glo@label</code> 67, 101, 406, 427, 657, 678	724, 743, 749, 750, 752, 753, 827
<code>\glo@list</code> 101, 427, 678	<code>\glossaryheader</code>
<code>\glo@name</code> 230, 502, 754	181, 221, <u>229</u> , 271, 275, 276,
<code>\glo@parent</code> 67, 406, 657	278–283, 285, 286, 289–294,
<code>\glo@type</code> 101, 427, 678	296–299, 301, 303, 304, 308,
<code>\glo@value</code> 86, 416, 667	310, 312, 315, 317–319, 321,
<code>\global</code> 17,	323, 472, 497, 502, 574, 593,
18, 81, 84, 94, 100, 105, 200,	595–600, 602–611, 613–615,
214, 222, 223, 227, 288, 378,	617–621, 723, 748, 754, 827,
379, 414, 416, 422, 426, 429,	846, 848–854, 856–865, 867–875
485, 491, 497, 498, 501, 602,	<code>\glossarymark</code> . . 51, <u>52</u> , 397, 647, 648
628, 629, 664, 666, 673, 677,	<code>\glossaryname</code> 19, <u>44</u> ,
680, 737, 743, 749, 750, 753, 855	46, 47, 379, 394, 395, 629, 644, 645
<code>\glo@linkprefix</code> . . . 123, <u>123</u> , 175,	<code>\glossarypostamble</code>
230, 442, 468, 502, 693, 720, 754 <u>51</u> , 181, 221, 397,
<code>\glosortentrieswarning</code> 23,	472, 497, 574, 647, 723, 749, 827
23, 215, 381, 382, 492, 631, 744	<code>\glossarypreamble</code> <u>50</u> , 181, 221, 397,
<code>\GlossariesWarning</code>	471, 496, 574, 647, 723, 748, 827
. . 7, 9, 23, <u>23</u> , 24, 27–29, 34–	<code>\glossarysection</code> <u>51</u> , 181, 221, 397,
36, 39, 51, 54, 66, 70, 71, 78,	471, 496, 574, 647, 723, 748, 827
81, 84, 85, 110, 111, 173, 188–	<code>\glossarystyle</code> <u>234</u> , 505, 757
191, 194, 196–198, 201, 202,	<code>glossarysubentry</code> (counter) <u>15</u>
208, 211, 212, 261, 333, 373,	<code>\glossarysubentryfield</code>
375, 381, 382, 384, 385, 388– <u>231</u> , 359, 503, 504,
391, 397, 399, 406, 408, 412,	557, 577–586, 755, 810, 830–839
414, 432, 433, 439, 440, 467,	

<code>\glossarytitle</code> . . .	181, 212, 213, 221, 225, 471, 491, 496, 499, 574, 723, 742, 743, 748, 751, 827	<code>\gls</code>	39, 110, <u>134</u> , 135, 194, 228, 239, 258, 391, 432, 450, 481, 501, 507, 519, 641, 683, 702, 733, 753, 759, 771
<code>\glossarytoctitle</code>	11, 19, 20, 40, 44, 47, 51, 181, 212, 221, 225, 226, 375, 379, 380, 391, 392, 394, 395, 397, 471, 491, 496, 499, 500, 574, 625, 629, 630, 642, 644, 645, 647, 723, 742, 748, 751, 752, 827	<code>\gls@delimN</code>	236
<code>\glossentry</code>	101, 213, 223, <u>230</u> , 271, 275–278, 280, 281, 290, 291, 293, 301, 303, 305, 308, 310, 312, 316, 317, 319, 321, 427, 491, 498, 503, 593, 595–598, 602–604, 609– 611, 613–615, 617–620, 678, 743, 750, 755, 846, 848–852, 856–858, 863–865, 867–872, 874	<code>\gls@accessibility</code>	333, 334, 536, 789, 790
<code>\Glossentrydesc</code> <u>231</u> , 359, 503, 556, 755, 809	<code>\gls@accsupp@engine</code>	333, <u>333</u> , 536, 789
<code>\glossentrydesc</code> . .	<u>230</u> , <u>231</u> , 272, 275–278, 280–282, 290, 291, 293, 301, 303, 305, 308–310, 312, 316, 317, 319, 321, 322, 359, 503, 556, 594–598, 602– 604, 609–611, 613–615, 617– 621, 754, 809, 847–852, 856– 858, 863–865, 867–869, 871–874	<code>\gls@Alphpage</code>	201, <u>202</u> , 206, 486, 487, 737–739
<code>\Glossentryname</code>	<u>230</u> , 502, 754	<code>\gls@alphpage</code>	201, <u>201</u> , 206, 486, 487, 737–739
<code>\glossentryname</code> 230, <u>230</u> , 232, 272, 274–278, 280, 281, 290, 291, 293, 301, 303, 305, 308, 310, 312, 316, 317, 319, 321, 322, 358, 359, 502, 556, 593–598, 602–604, 609–611, 613–615, 617–621, 754, 809, 847–852, 856–858, 863–865, 867–869, 871, 872, 874	<code>\gls@arabicpage</code>	201, <u>202</u> , 206, 486, 487, 737–739
<code>\Glossentrysymbol</code> <u>231</u> , 359, 503, 556, 755, 810	<code>\gls@assign@desc</code> 94, <u>94</u> , 99, 422, 426, 673, 677
<code>\glossentrysymbol</code>	231, <u>231</u> , 272, 281, 282, 293, 305, 312, 316, 317, 319, 321, 322, 359, 503, 556, 593, 594, 598, 604, 611, 615, 617–621, 755, 810, 847, 852, 858, 865, 869, 871, 872, 874	<code>\gls@assign@desc@field</code>	<u>24</u>
<code>\GLS</code>	<u>136</u> , 451, 702	<code>\gls@assign@descplural</code>	520, 527, 529, 531, 565, 568, 569, 772, 779, 780, 782, 784, 819, 821, 822
<code>\Gls</code> . .	110, 135, <u>135</u> , 239, 258, 432, 450, 507, 519, 683, 702, 759, 771	<code>\gls@assign@descplural@field</code> . .	<u>25</u>
		<code>\gls@assign@field</code>	83, <u>83</u> , 88, 89, 94, 96, 98–100, 326, 415, 418, 422, 423, 425, 426, 588, 665, 669, 673–677, 841
		<code>\gls@assign@firstpl</code>	520–523, 525, 527, 529–531, 565–569, 771, 772, 774–780, 782, 784, 818–822
		<code>\gls@assign@name@field</code>	<u>25</u>
		<code>\gls@assign@plural</code>	520–523, 525, 527, 529–531, 565–569, 772, 774–780, 782, 784, 819–822
		<code>\gls@assign@symbolplural</code> 520–523, 525, 529–531, 566, 567, 569, 772, 774–778, 782, 784, 819, 820, 822
		<code>\gls@assign@type@field</code>	<u>24</u>
		<code>\gls@automake@makegloss</code> 36, <u>36</u> , 37, 38, 192
		<code>\gls@automake@nr</code>	36, <u>36</u> , 37, 38, 191, 389, 480, 640, 731
		<code>\gls@automake@option</code>	36, <u>36</u> , 37
		<code>\gls@automake@val</code> . .	36, 37, 389, 640
		<code>\gls@begindocdefs</code> . .	85, <u>85</u> , 416, 667
		<code>\gls@checkisacronymlist</code> <u>22</u> , 121, 381, 441, 630, 692
		<code>\gls@checkseeallowed</code> 77, <u>77</u> , 193, 195, 412, 415, 480, 481, 662, 732, 733

<code>\gls@checkseeallowed@preambleonly</code>	<code>\gls@istfilebase</code> . . .	48, 48 , 189,
..... 78 , 412, 415, 662		190, 396, 478, 479, 646, 730, 731
<code>\gls@codepage</code>	<code>\gls@kernelglossredefs@nr</code>	33, 62, 189,
190, 215, 388, 404, 478, 479,	<code>\gls@kernelglossredefs@val</code>	39
492, 638, 654, 655, 730, 731, 744	<code>\gls@label</code>	238, 507, 759
<code>\gls@debug@nr</code> . .	<code>\gls@level</code>	81, 97, 98, 222,
<code>\gls@debug@val</code> .		414, 424, 497, 498, 664, 675, 749
<code>\gls@defdocnewglossaryentry</code> . . .	<code>\gls@mfirstuc@nr</code>	41
..... 83 , 85,	<code>\gls@mfirstuc@val</code>	41
107, 415, 416, 430, 666, 667, 681	<code>\gls@newacronym</code>	239, 239
<code>\gls@defglossaryentry</code> . . .	<code>\gls@noidx@nonumberlist</code> . . .	78 , 197
83–85,	<code>\gls@noidx@glossary</code>	
94, 95 , 415, 416, 422, 666, 667, 673	195, 198 , 481, 484, 733, 736
<code>\gls@delimN</code>	<code>\gls@nonumberlist@nr</code> . .	78, 412, 662
236, 236	<code>\gls@nonumberlist@val</code> .	78, 412, 662
<code>\gls@delimR</code>	<code>\gls@nosetquote</code>	
236, 236	95, 183, 184, 187, 422,
<code>\gls@disablepagerefexpansion</code> . .		473, 474, 477, 673, 724, 726, 729
200, 201 , 206, 485–487, 737–739	<code>\gls@number</code>	205, 487, 739
<code>\gls@do@addxdyattribute</code> 56, 400, 650	<code>\gls@numberedsection@nr</code>	
<code>\gls@docdef@addlabel</code>	10, 225, 375, 499, 500, 625, 751
.....	<code>\gls@numberedsection@val</code>	
84, 84 , 85, 666, 667	10, 225, 375, 499, 625, 751
<code>\gls@docdef@labellist</code> . .	<code>\gls@numberpage</code>	
84, 84 , 666	201, 202 , 205, 486, 487, 737–739	
<code>\gls@docclearpage</code>	<code>\gls@org@glossaryentryfield</code> . . .	
.....	213, 491, 743
53, 54, 398, 399, 648, 649	<code>\gls@org@glossarysubentryfield</code>	
<code>\gls@doentryfmt</code>	213, 214, 491, 743
71, 409, 659	<code>\gls@org@insert</code>	
<code>\gls@dosubst</code>	524, 526, 528, 776, 778, 779, 781	
126, 444, 694, 695	<code>\gls@orgAlph</code>	205, 206, 487, 739
<code>\gls@dotocitle</code>	<code>\gls@orgalph</code>	205, 206, 487, 739
212, 213, 225, 491, 499, 743, 751	<code>\gls@orgarabic</code> . . .	205, 206, 487, 739
<code>\gls@end@sanitizesort</code>	<code>\gls@orgnumber</code>	205, 487, 739
.....	<code>\gls@orgRoman</code>	205, 206, 487, 739
25, 26, 383, 632, 633	<code>\gls@orgromannumeral</code>	
<code>\gls@endcheck</code>	205, 206, 487, 739
82, 415, 665	<code>\gls@orgthe</code>	205, 487, 739
<code>\gls@glossary</code>	<code>\gls@original@glossary</code>	
.	38, 39, 390, 391, 640, 641
39, 40, 199 , 207, 208, 391,	<code>\gls@original@makeglossary</code>	
485, 488, 489, 641, 737, 740, 741	38, 40, 390, 391, 640, 641
<code>\gls@glsdefs@addlabel</code> 84 , 85, 666, 667	<code>\gls@protected@pagefmts</code> 125, 201,	
<code>\gls@glsdefs@labellist</code> .	201 , 202, 443, 486, 694, 737, 738	
84, 84 , 666	<code>\gls@Romanpage</code>	
<code>\gls@gobbleopt</code>	201, 202 , 206, 486, 487, 737–739	
.....	<code>\gls@romanpage</code>	
72, 73, 409, 410, 659, 660	201, 202 , 206, 486, 487, 737–739	
<code>\gls@grplabel</code>		
269, 592, 845		
<code>\gls@hypergroup prerun</code>		
.....		
269, 270 , 592, 845		
<code>\gls@ifnotmeasuring</code>		
.....		
103, 104, 428, 429, 679, 680		
<code>\gls@inlinepostchild</code>		
271–		
273, 577, 593, 594, 830, 846, 847		
<code>\gls@inlinesep</code>		
271,		
272, 577, 593, 594, 830, 846, 847		
<code>\gls@inlinesubsep</code>		
271,		
272, 577, 593, 594, 830, 846, 847		
<code>\gls@islistofacronyms</code>		
.....		
21, 380, 381, 630		

<code>\gls@save@numberlist</code>	11,	<code>\glsadd</code>	39, 175,
12, 210 , 375, 376, 490, 625, 741		175 , 176, 391, 468, 469, 641, 720	
<code>\gls@seenoindex@nr</code>	9, 374, 624	<code>\glsaddall</code>	176 , 469, 720
<code>\gls@seenoindex@val</code>	9, 374, 624	<code>\glsaddallunused</code>	176 , 469, 720
<code>\gls@set@xr@key</code>	77, 77 , 412, 662	<code>\glsaddkey</code>	88 , 418, 669
<code>\gls@suffixF</code>	49,	<code>\GlsAddLetterGroup</code>	63 , 405, 655
49 , 182, 184, 396, 472, 474,		<code>\glsaddprotectedpagefmt</code>	
574, 576, 646, 724–726, 828, 829		202, 202 , 486, 738
<code>\gls@suffixFF</code>	49,	<code>\GlsAddSortRule</code>	61 , 403, 404, 654
49 , 182, 184, 396, 472, 474,		<code>\glsaddstoragekey</code>	87 , 418, 668
574–576, 646, 724, 726, 828, 829		<code>\GlsAddXdyAlphabet</code>	57 , 401, 651
<code>\gls@text</code>	117, 439, 690	<code>\GlsAddXdyAttribute</code> 55 , 56 , 56 , 57 ,	
<code>\gls@the</code>	205, 487, 739	400 , 401, 571, 650, 651, 824, 825	
<code>\gls@thissty</code>	31, 387, 637	<code>\GlsAddXdyCounters</code>	
<code>\gls@tmp</code>	199, 260, 484, 528, 736, 781	55 , 55 , 56 , 57 , 74, 399,
<code>\gls@tmplen</code>	133, 320, 322, 449,	400 , 410, 571, 649–651, 660, 824	
583 , 619–621, 701, 836, 873, 874		<code>\GlsAddXdyLocation</code>	
<code>\gls@tr@set@acronym@toctitle</code>	59 , 402, 403, 571, 653, 825
.	20, 380, 629, 630	<code>\GlsAddXdyStyle</code>	61 , 404, 654
<code>\gls@tr@set@main@toctitle</code>		<code>\glsautomakefalse</code>	36 , 37 ,
.	19, 379, 629	191 , 192, 389, 390, 480, 640, 732	
<code>\gls@tr@set@numbers@toctitle</code> . .		<code>\glsautomaketrue</code>	36 , 37 , 389, 640
.	40, 392, 642	<code>\glsautoprefix</code>	10 ,
<code>\gls@tr@set@symbols@toctitle</code> . .		11 , 225, 226, 375, 500, 625, 752	
.	40, 391, 642	<code>\glsbackslash</code>	177 , 469, 721
<code>\gls@translate@nr</code>	30, 386, 636	<code>\glscapitalisewords</code>	169 , 169 , 717
<code>\gls@translate@val</code>	30, 386, 636	<code>\glscapscase</code>	
<code>\gls@wrglossary</code>	200, 200 , 485, 737	113 – 116 , 135 – 139 , 156 – 163 ,
<code>\gls@xdy@set@lang</code>	33 , 189, 190, 214	252 , 346 – 353 , 363 , 364 , 433 ,	
<code>\gls@xdystring</code>		434 , 436 – 438 , 450 – 453 , 459 –	
.	125 , 126 , 443 , 444 , 694, 695	463 , 515 , 524 , 528 , 544 , 545 ,	
<code>\gls@xindy@glslnumbersfalse</code>		547 – 552 , 560 , 684 , 685 , 687 –	
.	34 , 388, 638	689 , 702 – 704 , 710 – 715 , 767 ,	
<code>\gls@xindy@glslnumberstrue</code>		777 , 781, 797, 798, 800–806, 813	
.	33 , 388, 638	<code>\glscategory</code>	341 , 541, 794
<code>\gls@xr@key</code>	9,	<code>\glsclearpage</code> 53 , 54 , 398, 399, 648, 649	
77 , 77 , 78, 374, 375, 412, 624, 662		<code>\glsclosebrace</code>	60 ,
<code>\glsaccessdisplay</code>	345 , 544, 797	177 , 181, 182, 403, 469, 472,	
<code>\glsaccessibility</code>		574 , 575, 653, 721, 724, 827, 828	
.	333 , 342, 536, 541, 789, 794	<code>\glscompositor</code>	48 ,
<code>\glssacsupp</code> 341 , 342, 342 , 541, 794, 795		48 , 49, 59, 184, 396, 402, 474,	
<code>\glssacronymtrue</code>	20, 380, 630	574 , 576, 646, 652, 725, 827, 829	
<code>\glssacrpluralsuffix</code>		<code>\glscounter</code>	22 ,
.	45 , 45 , 240, 249, 250, 254,	22 , 43, 55, 73, 98, 123, 381,	
256 , 394, 507, 513, 516–518,		393 , 399, 410, 425, 442, 571,	
520 , 644, 759, 765, 768–770, 773		631 , 643, 649, 660, 676, 693, 824	
<code>\glssacrshortcutsfalse</code>	42 , 393, 643	<code>\glscurrententrylabel</code>	
<code>\glssacrshortcutstrue</code>	42 , 392, 643	210 , 213, 214, 490, 491, 742, 743	
<code>\glssacspace</code>	248 ,	<code>\glscurrentfieldvalue</code>	
248 , 250, 511, 512, 514, 763–765		69 , 70, 70 , 408, 409, 658, 659

<code>\glscustomtext</code> . . .	113, 115, 117, 135–139, 156–163, 252, 253, 346, 348, 350–353, 363, 364, 433, 436–438, 450–453, 459– 463, 515, 520, 521, 524, 526, 528, 544, 547, 549–552, 560, 684, 686–689, 702–704, 710– 715, 767, 773, 776–779, 781, 797, 800, 802, 804–806, 813, 814	464, 467, 468, 481–483, 487, 491, 493, 494, 497–499, 501, 502, 533–536, 542–544, 569, 570, 588, 620, 639, 656–658, 663, 667, 671–673, 677, 680– 682, 692, 715, 719, 720, 733– 735, 738, 739, 743, 745, 746, 749–751, 753, 754, 786–789, 795–797, 822, 823, 841, 842, 873	
<code>\GlsDeclareNoHyperList</code>	22, 23, 381, 631	<code>\glsdisablehyper</code>	134, 450, 701
<code>\glsdefaultshortaccess</code>	369, 369, 565–568, 818–821	<code>\Glsdisp</code>	140, 140
<code>\glsdefaulttype</code>	19, 19, 20, 33, 51, 62, 64, 71, 72, 96, 112, 199, 211, 212, 379, 397, 404, 405, 409, 423, 433, 439, 484, 490, 491, 629, 647, 654, 655, 659, 674, 684, 690, 736, 742	<code>\glsdisp</code>	139, 140, 453, 704
<code>\glsdefmain</code>	19, 20, 74, 379, 410, 629, 660	<code>\glsdisplay</code> 118, 433, 439, 440, 684, 690	
<code>\glsdefs@newdocentry</code> . . .	85, 86, 667	<code>\glsdisplayfirst</code>	117, 433, 439, 684, 690
<code>\GLSdesc</code>	146, 455, 707	<code>\glsdisplaynumberlist</code>	173, 196, 467, 482, 719, 734
<code>\Glsdesc</code>	146, 146, 455, 707	<code>\glsdohyperlink</code>	133, 133, 134, 449, 450, 701, 702
<code>\glsdesc</code>	145, 146, 455, 707	<code>\glsdohypertarget</code>	133, 133, 134, 449, 450, 701, 702
<code>\GLSdescplural</code>	147, 456, 707	<code>\glsdoifexists</code>	65, 67, 69, 91–94, 104, 135– 140, 156–163, 169, 173–175, 196, 197, 328–332, 351–353, 358, 359, 406–408, 420–422, 428, 429, 450–453, 459–463, 465, 467, 468, 482, 483, 551, 552, 556, 589–591, 656–658, 671–673, 679, 680, 702–705, 710–714, 717, 719, 720, 734, 735, 804, 805, 809, 810, 842–845
<code>\Glsdescplural</code> . . .	147, 147, 455, 707	<code>\glsdoifexistsordo</code>	66, 120, 164, 406, 441, 463, 656, 691, 715
<code>\glsdescplural</code> . . .	146, 147, 455, 707	<code>\glsdoifexistsorwarn</code>	66, 224, 230, 231, 406, 499, 502, 503, 656, 751, 754, 755
<code>\glsdescriptionaccessdisplay</code> . .	343, 355, 359, 542, 545–547, 554, 556, 557, 796, 799, 800, 807, 809, 810	<code>\glsdoifnoexists</code>	66, 83, 94, 406, 415, 422, 656, 665, 673
<code>\glsdescriptionpluralaccessdisplay</code>	344, 355, 542, 544, 545, 554, 796–798, 807	<code>\glsdonohyperlink</code>	123, 133, 133, 134, 442, 449, 450, 693, 701
<code>\glsdescwidth</code> .	278, 278, 279, 280, 283, 284, 286, 287, 289, 289, 290–294, 301, 301, 302–304, 306–308, 308, 309–314, 596– 599, 601–605, 609–616, 850, 851, 853–859, 862–864, 866–870	<code>\glsdoparenifnotempty</code> .	260, 528, 780
<code>\glsdetoklabel</code>	35, 65, 65, 66–70, 78, 79, 86, 91–95, 101, 105, 107–109, 122, 163, 164, 166, 174, 175, 195–197, 204, 206, 213, 214, 216–218, 222, 224, 225, 228, 230, 262–266, 320, 326–328, 342–345, 370, 371, 405–408, 413, 416, 420– 422, 427, 429–431, 441, 463,	<code>\glsdosanitizesort</code> .	16, 377, 378, 627
		<code>\glsdoshowtarget</code>	8, 133
		<code>\glsenableentrycount</code>	107, 430, 431, 681, 682
		<code>\glsenablehyper</code>	134, 450, 701
		<code>\glsencapwrcontent</code>	123, 123
		<code>\glsentryaccess</code>	338, 343, 540, 542, 793, 795

<code>\glentrycounter</code>	123 , 234 , 237 , 442 , 504 , 506 , 693 , 756 , 758	<code>\glentryfirstpluralaccess</code> 339 , 343 , 540 , 542 , 793 , 795
<code>\glentrycounterfalse</code> .	14 , 377 , 627	<code>\glentryfmt</code>	73 , 74 , 112 , 410 , 433 , 660 , 661 , 684
<code>\glentrycounterlabel</code> 228 , 229 , 501 , 502 , 753 , 754	<code>\Glsentryfull</code>	173 , 173 , 245 , 253 , 255 , 365 , 367 , 467 , 510 , 516 , 517 , 561 , 563 , 719 , 762 , 768 , 769 , 814 , 816
<code>\GlsEntryCounterLabelPrefix</code> 226 , 228 , 500 , 501 , 752 , 753	<code>\glentryfull</code>	173 , 173 , 245 , 253 , 255 , 365 , 367 , 467 , 510 , 516 , 517 , 561 , 563 , 718 , 762 , 768 , 769 , 814 , 816
<code>\glentrycountertrue</code> . .	14 , 377 , 627	<code>\Glsentryfullpl</code>	173 , 173 , 245 , 253 , 255 , 365 , 367 , 467 , 510 , 516 , 518 , 561 , 563 , 719 , 762 , 768 , 769 , 815 , 816
<code>\glentrycurrcount</code> 107 , 109 , 430–432 , 681–683	<code>\glentryfullpl</code>	173 , 173 , 245 , 253 , 255 , 365 , 367 , 467 , 510 , 516 , 517 , 561 , 563 , 718 , 762 , 768 , 769 , 814 , 816
<code>\Glsentrydesc</code>	146 , 167 , 167 , 231 , 355 , 359 , 455 , 464 , 503 , 554 , 556 , 707 , 716 , 755 , 807 , 809	<code>\glentryitem</code>	229 , 271 , 274–278 , 280 , 281 , 290 , 291 , 293 , 301 , 303 , 305 , 308 , 310 , 312 , 316 , 317 , 319 , 321 , 502 , 577–582 , 584–586 , 593 , 595–598 , 602–604 , 609–611 , 613–615 , 617–620 , 754 , 830– 839 , 846 , 848–852 , 856–858 , 863–865 , 867–869 , 871 , 872 , 874
<code>\glentrydesc</code> 146 , 167 , 167 , 230 , 355 , 359 , 434–436 , 455 , 464 , 503 , 545–547 , 554 , 556 , 685–687 , 707 , 716 , 754 , 799 , 800 , 807 , 809	<code>\Glsentrylong</code>	110 , 160 , 166 , 172 , 172 , 173 , 247 , 248 , 252 , 253 , 351 , 353 , 360 , 364 , 365 , 432 , 461 , 464 , 467 , 511 , 515 , 516 , 551 , 552 , 557 , 560 , 561 , 683 , 713 , 715 , 718 , 719 , 763 , 767 , 768 , 804 , 805 , 810 , 813 , 814
<code>\glentrydescaccess</code> 339 , 344 , 540 , 542 , 793 , 796	<code>\glentrylong</code>	110 , 117 , 160 , 161 , 166 , 172 , 172 , 173 , 247 , 248 , 250–256 , 350 , 352 , 353 , 360–369 , 432 , 439 , 461 , 462 , 464 , 467 , 511– 519 , 526 , 550 , 552 , 557–564 , 683 , 689 , 713 , 715 , 718 , 719 , 763–771 , 779 , 804–806 , 810–818
<code>\Glsentrydescplural</code> 147 , 167 , 167 ,	355 , 456 , 465 , 554 , 707 , 716 , 807	<code>\glentrylongaccess</code> 340 , 344 , 540 , 543 , 793 , 796
<code>\glentrydescplural</code>	147 , 167 , 167 , 355 , 433 , 434 , 455 , 456 , 464 , 544 , 545 , 554 , 684 , 685 , 707 , 716 , 797 , 798 , 807	<code>\Glsentrylongpl</code> 111 , 162 , 172 , 172 , 173 , 247 , 248 , 252 , 253 , 351 , 360 , 363 , 365 , 433 , 463 , 467 , 511 ,
<code>\glentrydescpluralaccess</code> 339 , 344 , 540 , 543 , 793 , 796		
<code>\Glsentryfirst</code> 110 , 114 , 142 , 168 ,	168 , 347 , 354 , 432 , 435 , 437 , 454 , 465 , 546 , 548 , 553 , 683 , 686 , 688 , 705 , 717 , 799 , 802 , 806		
<code>\glentryfirst</code>	110 , 114 , 142 , 168 , 168 , 347 , 348 , 354 , 432 , 434 , 435 , 437 , 454 , 465 , 546 , 548 , 549 , 553 , 683 , 685 , 686 , 688 , 705 , 717 , 799–802 , 806		
<code>\glentryfirstaccess</code> 339 , 343 , 540 , 542 , 793 , 795		
<code>\Glsentryfirstplural</code> 111 , 113 , 144 , 169 , 169 , 346 , 354 , 433 , 434 , 436 , 454 , 465 , 545 , 548 , 553 , 684 , 685 , 687 , 706 , 717 , 798 , 801 , 806		
<code>\glentryfirstplural</code> 111 , 113 , 144 , 169 , 169 , 346 , 347 , 354 , 432–434 , 436 , 454 , 455 , 465 , 544 , 545 , 547 , 548 , 553 , 683–685 , 687 , 706 , 717 , 797 , 798 , 800 , 801 , 806 , 807		

512, 515, 516, 551, 557, 560,	\glsentryprefixplural
561, 684, 714, 718, 719, 763, 326 , 327 , 329 ,
764, 767, 768, 804, 810, 813–815	332, 588, 589, 591, 841, 843, 845
\glsentrylongpl 111,	\glsentryprevcount
117, 162, 163, 172, 172 , 173, 107, 108, 430, 431, 681, 682
247–249, 252–255, 351, 360,	\Glsentryshort 115, 156, 166, 172,
361, 363, 365–367, 432, 439,	172 , 248, 254, 255, 349, 351,
462, 463, 467, 511, 512, 515–	361, 366, 367, 438, 459, 464,
518, 526, 532, 551, 557, 558,	466, 512, 517, 550, 551, 558,
560–563, 683, 690, 714, 715,	562, 563, 689, 711, 716, 718,
718, 719, 763, 764, 767–769,	764, 769, 803, 804, 811, 815, 816
779, 785, 804, 810, 811, 813–816	\glsentryshort 115–117, 156,
\glsentrylongpluralaccess	157, 166, 171 , 172, 173, 245,
. . . . 340 , 344, 540, 543, 794, 796	247–256, 349–352, 360–369,
\Glsentryname 145, 165, 165 ,	438, 439, 459, 460, 464, 466,
230, 355, 359, 455, 464, 502,	467, 510–519, 549–552, 557–
554, 556, 706, 715, 754, 807, 809	564, 689, 710, 711, 716, 718,
\glsentryname	719, 762–771, 803–805, 810–818
145, 165, 165 , 210, 320, 355,	\glsentryshortaccess
358, 455, 464, 553, 554, 556, 339 , 344, 540, 543, 793, 796
620, 706, 707, 715, 807, 809, 873	\Glsentryshortpl
\glsentrynumberlist 115, 158, 172, 172 , 249,
173 , 173 , 196, 467, 482, 719, 734	254, 255, 349, 361, 366, 367,
\glsentryparent 170 , 466, 717	437, 460, 467, 512, 517, 518,
\Glsentryplural 113,	549, 558, 562, 563, 688, 712,
143, 167 , 168, 346, 354, 434,	718, 764, 769, 802, 811, 815, 816
436, 454, 465, 544, 547, 553,	\glsentryshortpl 115,
684, 687, 706, 716, 798, 801, 806	117, 158, 159, 172, 172 , 173,
\glsentryplural 113, 143,	247, 248, 253–255, 348, 349,
167 , 168, 346, 347, 354, 433,	351, 360, 361, 365–367, 437,
434, 436, 454, 465, 544, 545,	439, 460, 461, 467, 511, 512,
547, 548, 553, 684, 685, 687,	516–518, 532, 549, 551, 557,
706, 716, 797, 798, 800, 801, 806	558, 561–563, 688, 690, 712,
\glsentrypluralaccess	718, 719, 763, 764, 768, 769,
. . . . 339 , 343, 540, 542, 793, 795	785, 802, 804, 810, 811, 814–816
\Glsentryprefix	\glsentryshortpluralaccess
327 , 327 , 330, 588, 590, 841, 843 339 , 344, 540, 543, 793, 796
\glsentryprefix . . . 326 , 327, 328,	\glsentrysort 170 , 466, 717
331, 588, 589, 591, 841, 842, 844	\Glsentrysymbol . . . 148, 168, 168 ,
\Glsentryprefixfirst	231, 356, 359, 456, 465, 503,
326 , 327, 330, 588, 590, 841, 843	554, 557, 708, 716, 755, 807, 810
\glsentryprefixfirst 326 , 327, 328,	\glsentrysymbol
331, 588, 589, 591, 841, 842, 844	148, 168, 168 , 231, 356, 359,
\Glsentryprefixfirstplural	434–436, 456, 465, 503, 521,
327 , 327 , 331, 588, 591, 841, 844	524, 528, 545–547, 554, 556,
\glsentryprefixfirstplural	685–687, 707, 708, 716, 755,
. 326 , 327, 329,	773, 777, 781, 799, 800, 807, 810
332, 588, 589, 592, 841, 843, 845	\glsentrysymbolaccess
\Glsentryprefixplural 339 , 343, 540, 542, 793, 795
327 , 327 , 330, 588, 590, 841, 844	

<code>\Glsentrysymbolplural</code>	<code>\Glsentryuseri</code>
. 149, 168, <u>168</u> ,	150, 170, <u>170</u> ,
356, 456, 465, 554, 708, 717, 808	356, 457, 466, 555, 708, 718, 808
<code>\glsentrysymbolplural</code>	<code>\glsentryuseri</code> 150, 170, <u>170</u> , 356,
. 149, 168, <u>168</u> , 356, 433,	456, 457, 466, 555, 708, 717, 808
434, 456, 465, 521, 524, 528,	<code>\glsentryuseriaccess</code>
544, 545, 554, 684, 685, 708, <u>340</u> , 344, 540, 543, 794, 796
717, 773, 777, 781, 797, 798, 808	<code>\Glsentryuserii</code> 151, 170, <u>170</u> ,
<code>\glsentrysymbolpluralaccess</code>	357, 457, 466, 555, 709, 718, 808
. <u>339</u> , 343, 540, 542, 793, 796	<code>\glsentryuserii</code> 151, 170, <u>170</u> , 357,
<code>\Glsentrytext</code> 114,	457, 466, 555, 708, 709, 718, 808
141, 167, <u>167</u> , 347, 353, 435,	<code>\glsentryuseriiaccess</code>
437, 453, 465, 546, 548, 553, <u>340</u> , 345, 540, 543, 794, 796
686, 688, 705, 716, 799, 801, 806	<code>\Glsentryuseriii</code> 152, <u>170</u> , 171,
<code>\glsentrytext</code>	357, 457, 466, 555, 709, 718, 808
114, 141, 167, <u>167</u> , 175, 210,	<code>\glsentryuseriii</code> 152, <u>170</u> , 171,
347, 348, 353, 434, 435, 437,	357, 457, 466, 555, 709, 718, 808
453, 465, 468, 490, 545, 546,	<code>\glsentryuseriiiaccess</code>
548, 549, 553, 685–688, 705, <u>340</u> , 345, 541, 543, 794, 796
716, 720, 741, 799, 801, 802, 806	<code>\Glsentryuseriv</code> 153, 171, <u>171</u> ,
<code>\glsentrytextaccess</code>	357, 458, 466, 555, 709, 718, 808
. <u>338</u> , 343, 540, 542, 793, 795	<code>\glsentryuseriv</code> 153, 171, <u>171</u> , 357,
<code>\glsentrytitlecase</code> <u>169</u> , 465, 466, 717	458, 466, 555, 709, 718, 808, 809
<code>\glsentrytype</code>	<code>\glsentryuserivaccess</code>
. 96, <u>169</u> , 423, 466, 674, 717 <u>340</u> , 345, 541, 543, 794, 797
<code>\Glsentryuseri</code> 150, 170, <u>170</u> ,	<code>\Glsentryuseriv</code> 154, 171, <u>171</u> ,
356, 457, 466, 555, 708, 718, 808	358, 458, 466, 555, 710, 718, 809
<code>\glsentryuseri</code> 150, 170, <u>170</u> , 356,	<code>\glsentryuseriv</code> 154, 171, <u>171</u> , 358,
456, 457, 466, 555, 708, 717, 808	458, 466, 555, 556, 710, 718, 809
<code>\glsentryuseriaccess</code>	<code>\glsentryuserivaccess</code>
. <u>340</u> , 344, 540, 543, 794, 796 <u>340</u> , 345, 541, 543, 794, 797
<code>\Glsentryuserii</code> 151, 170, <u>170</u> ,	<code>\Glsentryuserivi</code> 155, 171, <u>171</u> ,
357, 457, 466, 555, 709, 718, 808	358, 458, 466, 556, 710, 718, 809
<code>\glsentryuserii</code> 151, 170, <u>170</u> , 357,	<code>\glsentryuserivi</code> 155, 171, <u>171</u> , 358,
457, 466, 555, 708, 709, 718, 808	458, 459, 466, 556, 710, 718, 809
<code>\glsentryuseriiaccess</code>	<code>\glsentryuseriviaccess</code>
. <u>340</u> , 345, 540, 543, 794, 796 <u>340</u> , 345, 541, 544, 794, 797
<code>\Glsentryuseriii</code> 152, <u>170</u> , 171,	<code>\glsesclocationsfalse</code>
357, 457, 466, 555, 709, 718, 808 12, 195, 481, 733
<code>\glsentryuseriii</code> 152, <u>170</u> , 171,	<code>\glsesclocationstrue</code> 376, 625
357, 457, 466, 555, 709, 718, 808	<code>\glsexpandfields</code> <u>83</u> , 415, 665
<code>\glsentryuseriiiaccess</code>	<code>\glsfieldaccsupp</code>
. <u>340</u> , 345, 541, 543, 794, 796	341, <u>341</u> , 342, 541, 542, 794, 795
<code>\Glsentryuseriv</code> 153, 171, <u>171</u> ,	<code>\glsfielddef</code> <u>92</u> , 421, 671
357, 458, 466, 555, 709, 718, 808	<code>\glsfieldedef</code> <u>91</u> , 420, 671
<code>\glsentryuseriv</code> 153, 171, <u>171</u> , 357,	<code>\glsfieldfetch</code>
458, 466, 555, 709, 718, 808, 809 <u>92</u> , 169, 421, 465, 672, 717
<code>\glsentryuserivaccess</code>	<code>\glsfieldgdef</code> <u>91</u> , 420, 671
. <u>340</u> , 345, 541, 543, 794, 797	<code>\glsfieldxdef</code> <u>91</u> , 420, 671
<code>\Glsentryuseriv</code> 154, 171, <u>171</u> ,	<code>\glsfindwidesttoplevelname</code>
358, 458, 466, 555, 710, 718, 809 <u>320</u> , 619, 873
<code>\glsentryuseriv</code> 154, 171, <u>171</u> , 358,	<code>\GLSfirst</code> <u>142</u> , 454, 705
458, 466, 555, 556, 710, 718, 809	<code>\Glsfirst</code> 142, <u>142</u> , 454, 705
<code>\glsentryuserivaccess</code>	<code>\glsfirst</code> <u>141</u> , 142, 454, 705
. <u>340</u> , 345, 541, 543, 794, 797	<code>\glsfirstaccessdisplay</code>
<code>\Glsentryuservi</code> 155, 171, <u>171</u> ,	<u>343</u> , 347, 348, 354, 542, 546,
358, 458, 466, 556, 710, 718, 809	548, 549, 553, 795, 799–802, 806
<code>\glsentryuservi</code> 155, 171, <u>171</u> , 358,	<code>\GLSfirstplural</code> <u>144</u> , 454, 706
458, 459, 466, 556, 710, 718, 809	<code>\Glsfirstplural</code> 144, <u>144</u> , 454, 706
<code>\glsentryuserviaccess</code>	<code>\glsfirstplural</code> <u>143</u> , 144, 454, 706
. <u>340</u> , 345, 541, 544, 794, 797	<code>\glsfirstpluralaccessdisplay</code>
<code>\glsesclocationsfalse</code> <u>343</u> , 346, 354,
. 12, 195, 481, 733	542, 544, 545, 547, 548, 553,
<code>\glsesclocationstrue</code> 376, 625	795, 797, 798, 800, 801, 806, 807
<code>\glsexpandfields</code> <u>83</u> , 415, 665	<code>\glsfirstpluralacessdisplay</code>
<code>\glsfieldaccsupp</code> 347, 548, 801
341, <u>341</u> , 342, 541, 542, 794, 795	<code>\glsgenacfmt</code> <u>115</u> , 247, 248, 254,
<code>\glsfielddef</code> <u>92</u> , 421, 671	<u>348</u> , 360, 366, 437, 511, 512,
<code>\glsfieldedef</code> <u>91</u> , 420, 671	517, 549, 557, 558, 562, 688,
<code>\glsfieldfetch</code>	763, 764, 768, 802, 810, 811, 815
. <u>92</u> , 169, 421, 465, 672, 717	<code>\glsgenentryfmt</code> 112, <u>112</u> , 247, 248,
<code>\glsfieldgdef</code> <u>91</u> , 420, 671	253, 254, 258, <u>346</u> , 360, 364,
<code>\glsfieldxdef</code> <u>91</u> , 420, 671	366, 436, 511, 512, 515, 517,
<code>\glsfindwidesttoplevelname</code>	519, 521, 522, 524, 526, 528,
. <u>320</u> , 619, 873	530, 532, 547, 557, 558, 560,
<code>\GLSfirst</code> <u>142</u> , 454, 705	
<code>\Glsfirst</code> 142, <u>142</u> , 454, 705	
<code>\glsfirst</code> <u>141</u> , 142, 454, 705	
<code>\glsfirstaccessdisplay</code>	
<u>343</u> , 347, 348, 354, 542, 546,	
548, 549, 553, 795, 799–802, 806	
<code>\GLSfirstplural</code> <u>144</u> , 454, 706	
<code>\Glsfirstplural</code> 144, <u>144</u> , 454, 706	
<code>\glsfirstplural</code> <u>143</u> , 144, 454, 706	
<code>\glsfirstpluralaccessdisplay</code>	
. <u>343</u> , 346, 354,	
542, 544, 545, 547, 548, 553,	
795, 797, 798, 800, 801, 806, 807	
<code>\glsfirstpluralacessdisplay</code>	
. 347, 548, 801	
<code>\glsgenacfmt</code> <u>115</u> , 247, 248, 254,	
<u>348</u> , 360, 366, 437, 511, 512,	
517, 549, 557, 558, 562, 688,	
763, 764, 768, 802, 810, 811, 815	
<code>\glsgenentryfmt</code> 112, <u>112</u> , 247, 248,	
253, 254, 258, <u>346</u> , 360, 364,	
366, 436, 511, 512, 515, 517,	
519, 521, 522, 524, 526, 528,	
530, 532, 547, 557, 558, 560,	

562, 687, 763, 764, 767, 768,	\glsifusedtranslator	dict 29, 386, 636
771, 773, 775, 776, 779, 781,	\glsifusetranslator	29–31, 46, 47,
783, 785, 800, 810, 811, 814, 815		324, 386, 387, 394, 395, 570,
\glsgetgrouplabel		587, 635–637, 644, 645, 823, 840
\glsgetgrouptitle	\glsignore	176, 469, 720
274–276, 295–300, 316–320,	\glsindexingsetting .	15, 18, 192, 194
323, 504, 593, 595, 596, 606–	\glsindexonlyfirstfalse	31, 387, 637
609, 617–619, 621, 756, 846,	\glsinlinedescformat	
848, 849, 859–862, 871–873, 875		272, 273, 577, 594, 830, 847
\gls glossarymark . . .	\glsinlinedopostchild	271,
\gls groupheading		272, 577, 593, 594, 830, 846, 847
. . . . 182, 223, 232, 271, 275,	\glsinlineemptydescformat	
276, 278, 280, 281, 289, 291,		272, 273, 577, 593, 594, 830, 847
293, 295–301, 303, 305, 308,	\glsinlineifhaschildren . .	272, 272
310, 312, 315–321, 323, 472,	\glsinlinenameformat	
498, 504, 575, 593, 595–598,		272, 273, 577, 593, 594, 830, 847
602–604, 606–611, 613–615,	\glsinlineparentchildseparator	
617–621, 724, 750, 756, 828,		272, 273, 577, 594, 830, 847
846, 848–852, 856–865, 867–875	\glsinlinepostchild	
\gls groupskip		272, 273, 577, 594, 830, 847
. . . . 181, 182, 223, 232, 272,	\glsinlineseparator	
275, 279, 280, 282, 285, 286,		272, 273, 577, 594, 830, 847
290–293, 302, 303, 305, 309,	\glsinlinesubdescformat	
311, 313, 316, 318, 319, 322,		272, 273, 577, 594, 830, 847, 848
472, 498, 504, 574, 575, 594,	\glsinlinesubnameformat	
595, 597, 598, 600, 603, 604,		272, 273, 577, 594, 830, 847, 848
609–611, 613–615, 617–619,	\glsinlinesubseparator	
621, 724, 750, 756, 827, 828,		272, 273, 577, 594, 830, 847
847, 848, 850–854, 856–858,	\glsinsert . . .	113–116, 135–139,
863–865, 867–869, 871–873, 875		156–163, 252, 253, 346–353,
\glshyperfirstfalse		363, 364, 433–438, 450–453,
. . . . 254, 366, 517, 562, 769, 815		459–463, 515, 521, 524, 526,
\glshyperfirsttrue		528, 544–552, 560, 684–689,
\glshyperlink		702–704, 710–715, 767, 773,
. . . . 174, 210, 468, 490, 720, 741		776–779, 781, 797–806, 813, 814
\glshypernavsep 270, 270, 271, 593, 846	\glskeylisttok 244, 244, 245, 369,	
\glshypernumber . . .		509, 520–523, 525, 527, 529–
238, 396, 505–507, 647, 757–759		533, 565–569, 761, 771, 772,
\glsifhyper		774–780, 782–784, 786, 818–822
\glsifhyperon . 122, 440, 441, 691, 692	\glslabel . .	95, 113–116, 121–123,
\glsIfListOfAcronyms		156–163, 247, 248, 252–254,
. 21, 21, 22, 380, 381, 630		346–350, 360, 363, 364, 366,
\glsifplural 113, 115, 116, 135–139,		423, 433–438, 441, 442, 459–
156–163, 252, 346, 348, 349,		463, 511, 512, 515, 517, 521,
351–353, 363, 433, 436–438,		524, 526, 528, 544–550, 557,
450–453, 459–463, 515, 521,		558, 560, 562, 673, 684–689,
524, 526, 528, 544, 547, 549–		692, 693, 711–715, 763, 764,
552, 560, 684, 687–689, 702–		767, 768, 773, 776, 777, 779,
704, 710–715, 767, 773, 776,		781, 797–803, 810, 811, 813, 815
779, 781, 797, 800, 802–806, 813		

<code>\glslabeltok</code>	244, 244, 509, 519–533, 565–568, 761, 771–781, 783–786, 818–821	<code>\glsLTpenaltycheck</code>	288, 288, 601, 602, 855
<code>\glsletentryfield</code>	163, 463, 715	<code>\glsmakefirstuc</code>	41
<code>\Glslink</code>	121, 121	<code>\glsmcols</code>	295, 295, 296–300, 605–608, 859–862
<code>\glslink</code>	120, 121, 245, 253, 255, 259, 364–367, 440, 509, 510, 516, 517, 520, 561–563, 691, 761, 762, 767–769, 773, 814, 816	<code>\glsmfuaddmap</code>	5, 90, 110, 111, 121, 135, 138, 140–156, 158, 160, 162, 165, 167–173, 230, 231, 242, 243, 327, 329, 330
<code>\glslinkcheckfirsthyperhook</code>	121, 121, 441, 692	<code>\glsmfublocker</code>	5, 5, 90, 136, 138, 141–155, 157, 158, 160, 162
<code>\glslinkpostsetkeys</code>	122, 123, 441, 442, 692	<code>\glsmfuexcl</code>	5, 5, 175
<code>\glslinkvar</code>	119, 120, 440, 691	<code>\glsmoveentry</code>	101, 427, 677
<code>\glslistdottedwidth</code>	277, 277, 578, 596, 831, 849	<code>\GLSname</code>	145, 455, 706
<code>\glslistexpandedname</code>	274, 274, 848	<code>\Glsname</code>	145, 145, 455, 706
<code>\glslistgroupheaderfmt</code>	274, 275, 276, 595, 596, 848, 849	<code>\glsname</code>	144, 145, 455, 706
<code>\glslistinit</code>	274, 275, 848	<code>\glsnameaccessdisplay</code>	342, 355, 358, 359, 542, 553, 554, 556, 557, 795, 807, 809, 810
<code>\glslistnavigationitem</code>	274, 275, 276, 595, 848, 849	<code>\glsnamefont</code>	230, 235, 358, 359, 502, 503, 505, 536, 556, 754, 755, 757, 789, 809
<code>\glslocalreset</code>	104, 106, 428, 429, 679, 680	<code>\glsnavhyperlink</code>	268, 270, 271, 592, 593, 845, 846
<code>\glslocalresetall</code>	106, 429, 680	<code>\glsnavhyperlinkname</code>	269, 269, 592, 845
<code>\glslocalunset</code>	104, 106, 119, 429, 430, 450–453, 680, 681, 702–704	<code>\glsnavhypertarget</code>	269, 274–276, 296–300, 317, 318, 320, 323, 592, 595, 596, 606–609, 618, 619, 621, 845, 848, 849, 859–862, 871–873, 875
<code>\glslocalunsetall</code>	106, 430, 681	<code>\glsnavigation</code>	270, 274–276, 296– 300, 317–319, 323, 592, 595, 606–608, 618, 619, 621, 846, 848, 849, 859–862, 871–873, 875
<code>\glslocationcstoencap</code>	50	<code>\glsnextpages</code>	12, 78, 213, 227, 376, 412, 491, 501, 625, 662, 743, 753
<code>\glslongaccessdisplay</code>	344, 350– 353, 360–369, 543, 550–552, 557–565, 796, 804–806, 810–818	<code>\glsnoexpandfields</code>	83, 415, 665
<code>\glslongaccesskey</code>	369, 569, 822	<code>\glsnogroupskipfalse</code>	14, 376, 626
<code>\glslongkey</code>	240, 369, 507, 569, 759, 822	<code>\glsnoidxdisplayloc</code>	197, 198, 224, 483, 484, 499, 735, 736, 751
<code>\glslongpluralaccessdisplay</code>	344, 351, 360, 361, 363, 365–367, 543, 551, 557, 558, 560–563, 565, 796, 804, 810, 811, 813–816, 818	<code>\glsnoidxdisplaylocclishandler</code>	196, 224, 483, 498, 735, 750
<code>\glslongpluralaccesskey</code>	370, 569, 822	<code>\glsnoidxloclist</code>	196, 222, 223, 223, 482, 498, 734, 749, 750
<code>\glslongpluralkey</code>	240, 370, 507, 569, 759, 822	<code>\glsnoidxloclisthandler</code>	223, 223, 498, 750
<code>\glslongtok</code>	244, 244, 245, 247, 248, 253, 254, 360, 364, 366, 369, 509–512, 516, 517, 519–527, 529–533, 557, 558, 561, 562, 565–568, 761– 764, 767, 769, 771–780, 782– 786, 810, 811, 814, 815, 818–822	<code>\glsnoidxnumberlistloophandler</code>	197, 198, 483, 484, 735
<code>\glslowercase</code>	4	<code>\glsnoidxprenumberlist</code>	78, 222, 223

<code>\glsnoidxstripaccents</code>	<code>\glspl</code>
..... 26, 26, 383, 384, 633, 634	111, 137, 138,
<code>\glsnomakeindexwarning</code>	258, 432, 451, 519, 683, 703, 771
..... 54, 184, 399, 474, 649, 726	<code>\GLSplural</code>
<code>\glsnonextpages</code>	143, 454, 706
78, 213,	<code>\Glsplural</code>
227, 412, 491, 501, 662, 743, 753	143, 143, 454, 706
<code>\glsnopostdotfalse</code>	<code>\glsplural</code>
13, 376, 626	142, 143, 454, 706
<code>\glsnoxindywarning</code> ...	<code>\glspluralaccessdisplay</code>
49, 54, 55, 343, 346, 347,
57, 58, 60–63, 177, 178, 396,	354, 542, 544, 545, 547, 548,
399–401, 403, 404, 469, 470,	553, 795, 797, 798, 800, 801, 806
646, 649–651, 653–655, 721, 722	<code>\glspluralsuffix</code>
<code>\glsnumberformat</code>	45, 45, 98, 245, 247–249, 360,
..... 49, 174, 396, 467, 468, 647, 719	361, 365–368, 394, 425, 511,
<code>\glsnumberlistloop</code> 197, 197, 483, 735	512, 557, 558, 562, 564, 644,
<code>\glsnumbersgroupname</code>	675, 676, 763, 764, 811, 815–817
..... 40, 44, 47, 233, 392,	<code>\glspostdescription</code>
394, 395, 504, 642, 644, 646, 756	13,
<code>\glsnumlistlastsep</code>	47, 273, 275, 276, 278, 290,
174, 174, 196, 468, 483, 720, 734	301, 308, 309, 316, 317, 319,
<code>\glsnumlistparser</code>	321, 322, 376, 396, 577–585,
..... 174, 193, 468, 481, 720, 732	594–597, 602, 609, 613, 617–
<code>\glsnumlistsep</code>	621, 626, 646, 831, 833–839,
174, 174, 196, 468, 483, 720, 734	847–850, 856, 863, 867, 871–874
<code>\glsopenbrace</code>	<code>\glspostinline</code>
60, 271, 273, 593, 594, 846, 847
177, 181, 182, 403, 469, 472,	<code>\glspostlinkhook</code> .
574, 575, 653, 720, 724, 827, 828	121, 121, 135–
<code>\glsorder</code>	140, 156–163, 351–353, 441,
32, 32,	450–453, 459–463, 551–553,
189–191, 193, 219, 388, 478–	692, 702–705, 711–715, 804–806
480, 495, 637, 638, 729–732, 747	<code>\glsprefixsep</code>
<code>\glsorg@endtheglossary</code> 6, 6, 372, 622	328,
<code>\glsorg@PrintChanges</code> ...	328, 329–332, 589–592, 842–845
6, 372, 622	<code>\glsprestandardsort</code>
<code>\glsorg@theglossary</code> 15, 16, 377, 378, 627
6, 6, 372, 622	<code>\glsquote</code>
<code>\glspagelistwidth</code>	177, 469, 721
278, 280, 283, 284, 286, 287,	<code>\glsrefentry</code>
289, 291–294, 301, 303, 304,	228, 501, 753
306, 307, 308, 310–314, 596–	<code>\glsreset</code> 104, 105, 428, 429, 679, 680
599, 601–605, 609–616, 850,	<code>\glsresetall</code>
851, 853–859, 863, 864, 866–870	105, 429, 680
<code>\glspar</code>	<code>\glsresetcurrcountfalse</code>
47, 396, 646	107
<code>\glspatchLToutput</code>	<code>\glsresetentrycounter</code> .
284–287, 288, 600, 601, 853–855	228, 501, 753
<code>\glspatchtabularx</code>	<code>\glsresetentrylist</code>
103, 428, 679	181,
<code>\glspenaltygroupskip</code>	221, 227, 227, 472, 497, 500,
285, 286, 288, 600, 601, 853–855	501, 574, 724, 748, 752, 753, 827
<code>\glspercentchar</code>	<code>\glsresetsubentrycounter</code>
86, 228, 229, 272, 501,
177, 181, 183, 416, 417, 469,	502, 577, 594, 753, 754, 830, 847
472, 473, 667, 668, 721, 723, 725	<code>\glsrestoreLToutput</code> ...
<code>\GLSpl</code>	288, 601, 855
138, 452, 704	<code>\glssanitizesortfalse</code> .
<code>\Glspl</code>	28, 385, 635
111, 137, 138,	<code>\glssanitizesorttrue</code> 28, 385, 634, 635
258, 433, 452, 519, 683, 703, 771	<code>\glssavenumberlistfalse</code> 12, 376, 625
	<code>\glssavewritesfalse</code> ...
	38, 390, 640
	<code>\glssee</code>
	209, 489, 741

<code>\glseeformat</code>	180,	<code>\glsshortpluralaccesskey</code>	
195, 197, 209, 471, 482, 483,		369, 569, 822	
489, 574, 723, 733, 735, 741, 827		<code>\glsshortpluralkey</code>	
<code>\glseeitem</code>	209, 210, 489, 490, 741	240, 369, 507, 569, 759, 822	
<code>\glseeitemformat</code>	210, 210, 490, 741	<code>\glsshorttok</code>	244, 244, 245,
<code>\glseelastsep</code>	209, 210, 489, 741	369, 509, 519–527, 529–533,	
<code>\glseelist</code>	209, 209, 489, 741	565–569, 761, 771–786, 818–822	
<code>\glseesep</code>	209, 210, 489, 741	<code>\glsshowaccsupp</code> 7, 9, 373, 374, 623, 624	
<code>\glssentencecase</code>		<code>\glsshowtarget</code>	7, 8, 373, 374, 623
4, 117, 121, 140, 164–166		<code>\glsshowtargetfont</code> 8, 8, 373, 374, 623	
<code>\glsetAlphaCompositor</code>	49, 396, 646	<code>\glsshowtargetfonttext</code>	8, 8
<code>\glsetCompositor</code>	48, 396, 646	<code>\glsshowtargetinner</code>	8, 8
<code>\glsetexpandfield</code>	24, 24,	<code>\glsshowtargetouter</code>	8, 8, 374, 623
27–29, 382, 384, 385, 632, 634, 635		<code>\glsshowtargetsymbols</code>	8, 8, 374, 623
<code>\glsetnoexpandfield</code>	24, 24, 25,	<code>\glssortnumberfmt</code>	
27, 28, 382, 384, 385, 632, 634, 635		16, 17, 18, 378, 379, 627, 628	
<code>\GlsSetQuote</code>	95, 183, 184, 422,	<code>\glsspace</code>	241, 241, 508, 759, 772
473, 474, 477, 673, 724, 726, 729		<code>\glststepentry</code>	
<code>\glsetSuffixF</code>	49, 396, 646	228, 229, 501, 502, 753, 754	
<code>\glsetSuffixFF</code>	49, 396, 646	<code>\glststepsubentry</code>	
<code>\glsettoctitle</code>	44, 47,	228, 229, 501, 502, 753, 754	
212, 394, 395, 491, 644, 645, 743		<code>\glssubentrycounterfalse</code> 15, 377, 627	
<code>\glsetwidest</code>	320, 620, 873	<code>\glssubentrycounterlabel</code>	
<code>\GlsSetWriteIstHook</code>	184, 474, 726	229, 229, 501, 502, 753, 754	
<code>\GlsSetXdyCodePage</code>	62, 404, 654, 655	<code>\glssubentryitem</code>	229, 272, 275–
<code>\GlsSetXdyFirstLetterAfterDigits</code>		278, 280, 282, 290, 291, 293,	
177, 469, 721		301, 303, 305, 309, 310, 312,	
<code>\GlsSetXdyLanguage</code>	62, 404, 654	316, 317, 319, 321, 502, 577–	
<code>\GlsSetXdyLocationClassOrder</code> . .		586, 594–598, 602–604, 609–	
61, 403, 653		611, 613–615, 617–620, 754,	
<code>\GlsSetXdyMinRangeLength</code>		830–839, 847–852, 856–858,	
178, 470, 721, 722		863–865, 867–869, 871, 872, 874	
<code>\GlsSetXdyNumberGroupOrder</code>		<code>\GLSsymbol</code>	148, 456, 708
178, 469, 470, 721		<code>\Glsymbol</code>	148, 148, 456, 707
<code>\GlsSetXdyStyles</code>	61, 404, 654	<code>\glssymbol</code>	147, 148, 456, 707
<code>\glsshortaccessdisplay</code>		<code>\glssymbolaccessdisplay</code>	343,
166, 344, 349–352, 360–		356, 359, 542, 545–547, 554,	
369, 464, 543, 549–552, 557–		556, 557, 795, 799, 800, 807, 810	
565, 716, 796, 803–805, 810–818		<code>\glssymbolnav</code>	271, 593, 846
<code>\glsshortaccesskey</code>	369, 569, 822	<code>\GLSsymbolplural</code>	149, 456, 708
<code>\glsshortaccsupp</code>	342, 342, 541, 794	<code>\Glsymbolplural</code>	149, 149, 456, 708
<code>\glsshortkey</code>		<code>\glssymbolplural</code>	148, 149, 456, 708
240, 369, 507, 569, 759, 822		<code>\glssymbolpluralaccessdisplay</code> . .	
<code>\glsshortplaccsupp</code>	342, 541, 794	343, 356, 542,	
<code>\glsshortpluralaccessdisplay</code> . .		544, 545, 554, 795, 797, 798, 808	
344, 348, 349, 351, 360,		<code>\glssymbolsgroupname</code>	
361, 365–367, 543, 549, 551,		40, 44, 47, 233, 391,	
557, 558, 561–563, 565, 796,		394, 395, 504, 642, 644, 646, 756	
802, 804, 810, 811, 814–816, 818		<code>\glstarget</code>	230, 273–278, 280–
		282, 290, 291, 293, 301, 303,	

305, 308–310, 312, 316, 317, 319, 321, 322, 502–504, 577– 586, 594–598, 602–604, 609– 611, 613–615, 617–621, 754– 756, 830–839, 847–852, 856– 858, 863–865, 867–869, 871–874	\glstypexorpdfstring
164, 164, 165, 169, 203	121, 122, 135–
\GLStext 141, 453, 705	139, 156–163, 351–353, 441, 450–453, 459–463, 551, 552, 692, 702–704, 710–715, 804–806
\GlStext 141, 141, 453, 705	\glsucmarkfalse 14, 376, 626
\glstext 140, 141, 453, 705	\glsucmarktrue 14, 376, 626
\glstextaccessdisplay . 343, 347, 348, 353, 542, 545, 546, 548, 549, 553, 795, 799, 801, 802, 806	\glsunexpandedfieldvalue 92, 274, 672, 848
\glstextformat 112, 121, 123, 433, 441, 442, 684, 692, 693	\glsunset 103, 104, 106, 108, 119, 428– 431, 450–453, 679–682, 702–705
\glstextup 45, 240, 367, 394, 507, 564, 644, 759, 817	\glsunsetall 106, 430, 680
\glstildechar . . 56, 177, 181, 182, 400, 469, 472, 650, 721, 723, 724	\glsupacrpluralsuffix 45, 249, 255, 394, 513, 518, 522, 525, 527, 530, 644, 765, 770, 775, 778, 780, 783
\glstranslatefalse 30, 31, 386, 387, 636	\glsupercase . 4, 52, 90, 113–116, 141–155, 157, 159, 161, 163, 245, 252, 253, 255, 331, 332, 347–350, 352–358, 363–365, 367
\glstranslatetrue 30, 31, 386, 387, 636, 637	\GlsUseAcrEntryDispStyle . . 246, 247, 249–251, 254–256, 361– 363, 365, 367, 368, 510–516, 518, 558, 559, 562–564, 762– 766, 768, 770, 811–813, 815, 817
\glstreechildpredesc 315, 316, 317, 617, 618, 870–872	\GlsUseAcrStyleDefs 246, 247, 249–252, 254–256, 361–363, 365, 367– 369, 510, 511, 513–516, 518, 519, 558, 559, 562–564, 762– 768, 770, 811–813, 815, 817, 818
\glstreegroupheaderfmt 295–300, 314, 316–320, 323, 606–609, 616– 619, 621, 859–862, 870–873, 875	\GLSuseri 150, 457, 708
\glstreeindent 317, 318, 319, 321, 322, 581– 583, 618–621, 835, 836, 872, 874	\Glsuseri 150, 150, 457, 708
\glstreeitem . . 295, 296, 315, 315, 605, 606, 616, 617, 859, 860, 870	\glsuseri 149, 150, 456, 708
\glstreenamebox 320, 321, 322, 620, 621, 873, 874	\glsuseriaccessdisplay 344, 356, 543, 555, 796, 808
\glstreenamefmt . . 314, 314, 316, 317, 319–322, 616–621, 870–874	\GLSuserii 151, 457, 709
\glstreenavigationfmt 296–300, 314, 317– 319, 323, 606–608, 616, 618, 619, 621, 859–862, 870–873, 875	\Glsuserii 151, 151, 457, 709
\glstreepredesc 315, 316, 317, 319, 617–619, 870–872	\glsuserii 150, 151, 457, 708
\glstreesubitem 295, 315, 315, 605, 617, 859, 870	\glsuseriiaccessdisplay 344, 357, 543, 555, 796, 808
\glstreesubsubitem 295, 315, 315, 605, 617, 859, 870	\GLSuseriii 152, 457, 709
	\Glsuseriii 152, 152, 457, 709
	\glsuseriii 151, 152, 457, 709
	\glsuseriiiaccessdisplay 345, 357, 543, 555, 796, 808
	\GLSuseriv 153, 458, 709
	\Glsuseriv 153, 153, 458, 709
	\glsuseriv 152, 153, 457, 709

<code>\ifcsequal</code>	68, 407, 657	731, 733, 746, 747, 749, 750, 789
<code>\ifcsname</code>	93, 672	
<code>\ifcsstrequal</code>	94, 422, 673	189, 190, 222, 223, 383, 426,
<code>\ifcsstring</code>	93, 421, 672	497, 498, 632, 633, 677, 749, 750
<code>\ifcsundef</code>	10, 17, 33, 43, 46,	
	50, 51, 53, 64, 65, 73, 74, 77,	<code>\ifdim</code>
	96, 98, 99, 107, 118, 123, 124,	248, 288,
	133, 199, 211, 216, 224, 225,	320, 512, 601, 620, 764, 855, 873
	229, 233–235, 238, 246, 270,	<code>\iffalse</code>
	342–345, 375, 378, 388, 393,	100, 105, 426, 429, 677, 680
	394, 396–398, 405, 406, 409–	<code>\IfFileExists</code>
	411, 423, 425, 430, 440, 442,	12, 29, 30, 190–
	449, 450, 478, 479, 484, 490,	192, 214, 376, 386, 478, 479,
	492, 493, 499, 502, 504, 505,	492, 626, 635, 636, 730, 731, 743
	507, 510, 542–544, 576, 592,	<code>\ifglossaryexists</code>
	624, 628, 638, 643, 645, 647,	51, 62,
	648, 656, 660, 662, 674, 676,	<u>64</u> , 66, 188–191, 397, 404–406,
	681, 691, 693, 701, 730, 736,	477–479, 647, 654–656, 729–731
	742, 744, 745, 751, 754, 756,	<code>\ifglsc@sanitize@description</code>
	757, 759, 762, 795–797, 830, 846	27, 384, 634
<code>\ifcsvoid</code>	69	<code>\ifglsc@sanitize@name</code>
<code>\ifdef</code> 4, 42, 69, 79, 85, 86, 103, 119,		27, 384, 634
164, 196, 197, 203, 240, 274,		<code>\ifglsc@sanitize@symbol</code>
315, 328, 341, 408, 413, 416,		28, 384, 634
428, 440, 463, 465, 482, 483,		<code>\ifglsc@xindy@glscnumbers</code>
507, 541, 589, 616, 617, 658,		63, 405, 655
663, 667, 679, 691, 715, 717,		<code>\ifglsc@description</code>
734, 735, 759, 794, 842, 848, 870		32, 531, 785
<code>\ifdefempty</code>	22, 35, 52, 53, 64,	<code>\ifglsc@rdua</code>
68–70, 74, 113, 115, 191, 192,		32, 522,
194, 221, 223, 244, 246, 252,		527, 530–532, 775, 780, 783, 785
260, 346, 348, 363, 381, 389,		<code>\ifglsc@footnote</code>
398, 405, 407, 408, 410, 433,	
436, 437, 480, 481, 497, 498,		31, 121, 441, 531, 532, 692, 785
509, 510, 515, 520, 524, 526,		<code>\ifglsc@acronym</code>
528, 544, 547, 549, 560, 631,		20, 379, 380, 629
639, 648, 655, 657, 658, 660,		<code>\ifglsc@shortcuts</code>
661, 684, 687, 688, 732, 733,	
749, 750, 761, 762, 767, 773,		42, 258, 392, 519, 643, 771
776, 778, 781, 797, 800, 802, 813		<code>\ifglsc@smallcaps</code>
<code>\ifdefequal</code> 34, 67–70, 81, 82, 87, 97,	
101, 223, 342, 388, 389, 406–		32, 522, 523, 525, 527,
408, 414, 415, 417, 424, 427,		530, 774–776, 778, 780, 782, 783
498, 542, 638, 639, 657, 658,		<code>\ifglsc@smaller</code>
664, 665, 668, 675, 678, 750, 795		32, 522,
<code>\ifdefstrequal</code>	93, 421, 672	523, 526, 527, 774–776, 778, 780
<code>\ifdefstring</code>	13, 46, 71, 189–191,	<code>\ifglsc@automake</code>
194, 218, 219, 222–224, 333,		36, 37, 42,
376, 394, 409, 478–480, 495–		193, 389, 390, 481, 639, 640, 732
499, 536, 626, 644, 659, 729–		<code>\ifglsc@descsuppressed</code>
	
		68, 272, 407, 593, 657, 847
		<code>\ifglsc@entrycounter</code>
		14, 15, 42, 228, 229, 377, 393,
		501, 502, 626, 627, 643, 753, 754
		<code>\ifglsc@entryexists</code>
		65, <u>65</u> , 66,
		83–85, 95, 97, 405, 406, 416,
		422, 424, 656, 666, 667, 673, 675
		<code>\ifglsc@sclocations</code>
	
		125, 201–203, 486, 738
		<code>\ifglsc@fieldcseq</code>
		94, 422, 673
		<code>\ifglsc@fielddefeq</code>
		93, 421, 672
		<code>\ifglsc@fieldeq</code>
		93, 421, 672
		<code>\ifglsc@fieldvoid</code>
		<u>68</u>
		<code>\ifglsc@haschildren</code>
		<u>67</u> ,
		272, 406, 577, 594, 656, 830, 847

<code>\ifglshasdesc</code>	<code>\ifglstoc</code>
..... 67, 272, 407, 594, 657, 847	54, 399, 649
<code>\ifglshasfield</code>	<code>\ifglstranslate</code>
69, 408, 658 29, 45, 386, 394, 635, 644
<code>\ifglshaslong</code>	<code>\ifglsucmark</code>
68,	52, 397, 647, 648
110, 111, 165, 166, 247, 248,	<code>\ifglssused</code>
252, 254, 360, 363, 366, 407,	65, 109, 113–
432, 433, 464, 511, 512, 515,	115, 121, 176, 201, 328–332,
517, 526, 557, 558, 560, 562,	346–348, 406, 432–437, 441,
657, 683, 684, 715, 763, 764,	469, 485, 521, 524, 526, 528,
767, 768, 779, 810, 811, 813, 815	544–549, 589–591, 656, 682,
<code>\ifglshasparent</code>	684–688, 692, 720, 737, 773,
..... 67, 216, 222, 320, 407,	776, 779, 781, 797–802, 842–845
493, 497, 619, 657, 745, 749, 873	<code>\ifglswrallowprimitivemods</code>
<code>\ifglshasprefix</code> 204, 205, 487, 739
327,	<code>\ifglsxindy</code>
328, 330, 331, 588–591, 841–844	15, 32, 48,
<code>\ifglshasprefixfirst</code>	49, 54–63, 74, 101, 102, 126,
327,	177, 178, 184, 189, 190, 192,
328, 330, 331, 588–591, 842–844	206, 208, 214, 388, 396, 399–
<code>\ifglshasprefixfirstplural</code>	405, 410, 427, 444, 469, 470,
..... 328, 329, 331,	474, 477, 478, 488, 489, 492,
332, 588, 589, 591, 592, 842–845	571, 572, 638, 646, 649–655,
<code>\ifglshasprefixplural</code> .	660, 678, 695, 721, 722, 726,
327, 329,	729, 730, 739, 741, 744, 824–826
330, 332, 588–591, 841, 843–845	<code>\ifignoredglossary</code>
<code>\ifglshasshort</code>	75, 96,
68, 210, 407, 658	100, 201, 211, 411, 423, 426,
<code>\ifglshassymbol</code>	485, 490, 661, 674, 677, 737, 742
. 68, 316, 317, 319, 321, 322,	<code>\ifin@</code>
407, 521, 524, 528, 617–621,	41, 392, 642
657, 773, 776, 781, 871, 872, 874	<code>\ifinlistcs</code> 220, 224, 496, 499, 748, 751
<code>\ifglshyperfirst</code>	<code>\ifinner</code>
121, 441, 692	8, 373, 623
<code>\ifglsindeonlyfirst</code> ..	<code>\ifKV@glslink@hyper</code>
201, 485, 737 122, 123, 441, 442, 692, 693
<code>\ifglsnogroupskip</code>	<code>\ifKV@glslink@local</code>
275, 119, 450–453, 702–704
279, 280, 282, 285, 286, 290,	<code>\ifmeasuring@</code>
291, 293, 301, 303, 305, 309,	103, 428, 679
311, 313, 316, 318, 319, 322,	<code>\ifmmode</code>
595, 597, 598, 600, 603, 604,	8, 373, 623
609–611, 613–615, 617–619,	<code>\ifnum</code> .
621, 848, 850–854, 856–858,	16, 108, 191, 222, 288, 317,
863–865, 867–869, 871–873, 875	319, 321, 322, 378, 389, 431,
<code>\ifglsnonumberlist</code>	480, 497, 581–583, 601, 618–
226, 500, 752	620, 627, 628, 640, 681, 682,
<code>\ifglsnopostdot</code>	731, 749, 835, 836, 855, 872, 874
13, 376, 626	<code>\ifstreempty</code>
<code>\ifglsnnumberline</code>	9, 374, 577, 624, 830
54, 399, 649	<code>\ifstrequal</code>
<code>\ifglsrresetcurrcount</code>	504, 756
106, 107	<code>\ifthenelse</code>
<code>\ifglssanitizesort</code>	29, 43, 53,
.. 25, 28, 382, 385, 632, 634, 635	123, 193, 269, 385, 393, 398,
<code>\ifglssavenumberlist</code> ...	442, 480, 504, 532, 592, 635,
81, 193,	643, 648, 693, 732, 756, 785, 845
210, 414, 481, 490, 664, 732, 741	<code>\IfTrackedLanguage</code>
<code>\ifglssavewrites</code>	185, 474, 726
38, 188,	<code>\IfTrackedLanguageFileExists</code> ..
200, 390, 477, 485, 640, 729, 737 46, 324,
<code>\ifglssubentrycounter</code>	325, 395, 570, 587, 645, 823, 840
..... 15, 42, 228, 229, 377,	<code>\iftrue</code> ..
393, 501, 502, 627, 643, 753, 754	100, 105, 426, 429, 677, 680

<code>\ifundef</code>	14, 15, 72, 84, 96, 179, 183, 192, 377, 409, 416, 423, 470, 473, 480, 626, 627, 659, 666, 674, 722, 725, 732	470, 473, 478–480, 572, 575, 646, 722, 725, 730–732, 826, 828
<code>\ifvmode</code>	176, 468, 720	
<code>\ifvoid</code>	288, 602, 855	
<code>\ifx</code>	14, 16, 18, 21, 35, 40, 41, 43, 55, 56, 58, 60, 62, 63, 71, 97–99, 102, 123, 124, 127–132, 166, 179, 182, 184, 185, 187, 199, 203, 205, 206, 208, 209, 212, 213, 234, 236, 237, 260, 262, 333, 342, 377–381, 392, 393, 399, 400, 402, 404, 405, 423–427, 442, 444–449, 464, 470, 472, 474–476, 484, 486– 489, 491, 504–506, 520, 522, 523, 525, 527–529, 531, 533, 535, 541, 572–577, 581–583, 626–628, 630, 639, 642, 643, 649, 650, 652, 653, 655, 674– 678, 693, 695–701, 715, 716, 722, 724, 726–728, 736, 738– 741, 743, 756–758, 772, 774, 776, 778, 780–782, 784, 786, 789, 795, 826, 828–830, 834–836	<code>\item</code>
<code>\immediate</code>	35, 84, 86, 109, 189–192, 199, 214, 215, 389, 416, 417, 432, 477, 479, 485, 492, 639, 666, 667, 682, 683, 729, 731, 736, 737, 744	274–277, 295, 296, 315–317, 577, 578, 580, 581, 595, 596, 605, 606, 617, 618, 830, 831, 834, 848–850, 859, 860, 870, 871
<code>\in@</code>	41, 392, 642	
<code>index</code> (option)	40	
<code>index</code> (style)	315	
<code>indexgroup</code> (style)	316	
<code>indexhypergroup</code> (style)	316	
<code>\indexname</code>	41, 392, 642	
<code>indexonlyfirst</code> (option)	31	
<code>\indexspace</code> 274, 275, 295, 295, 296– 300, 314, 316–320, 322, 323, 594, 595, 605–609, 616–619, 621, 848, 859–862, 870–873, 875		
<code>inline</code> (style)	271	
<code>\input</code>	45, 112, 394, 433, 644, 684	
<code>\inputencodingname</code>	33, 388, 638	
<code>\InputIfFileExists</code>	85, 416, 667	
<code>\istfile</code>	198, 484, 736	
<code>\istfilename</code>	48, 48, 179, 183, 190, 191, 193, 396,	
J		
<code>\jobname</code>	35, 48, 84, 85, 179, 183, 189–192, 214, 389, 396, 416, 470, 473, 477–479, 492, 572, 575, 639, 646, 666, 667, 722, 725, 729–731, 743, 826, 828	
K		
<code>kernelglossredefs</code> (option)	39	
<code>\key@ifundefined</code> 87–89, 418, 668, 669		
keys:		
<code>access</code>	334	
<code>counter</code>	77	
<code>description</code>	75	
<code>descriptionaccess</code>	335	
<code>descriptionplural</code>	75	
<code>descriptionpluralaccess</code>	335	
<code>first</code>	76	
<code>firstaccess</code>	334	
<code>firstplural</code>	76	
<code>firstpluralaccess</code>	334	
<code>long</code>	80	
<code>longaccess</code>	335	
<code>longplural</code>	80	
<code>longpluralaccess</code>	335	
<code>name</code>	75	
<code>nonumberlist</code>	78	
<code>parent</code>	78	
<code>plural</code>	76	
<code>pluralaccess</code>	334	
<code>see</code>	77	
<code>short</code>	80	
<code>shortaccess</code>	335	
<code>shortplural</code>	80	
<code>shortpluralaccess</code>	335	
<code>sort</code>	76	
<code>symbol</code>	76	
<code>symbolaccess</code>	334	
<code>symbolplural</code>	77	
<code>symbolpluralaccess</code>	334	
<code>text</code>	76	
<code>textaccess</code>	334	
<code>type</code>	77	

user1	79	639–646, 651, 658, 659, 664–
user1access	335	666, 673–677, 679–682, 684,
user2	79	691–695, 701–705, 710–716,
user2access	335	719, 720, 724, 726, 729–735,
user3	79	737–739, 741–743, 750, 751,
user3access	335	753, 755, 761, 771, 772, 774–
user4	79	782, 784, 791, 792, 804–806,
user4access	336	818–823, 829, 840, 841, 845,
user5	79	846, 848, 855, 859, 860, 870
user5access	336	\letcs
user6	80	33, 35, 67–
user6access	336	70, 86, 88, 89, 92, 98, 99, 164,
\KV@glslink@hyperfalse		166, 196, 197, 216–218, 222,
	119, 121, 122,	230, 233, 270, 320, 406–408,
	134, 440, 441, 450, 691, 692, 701	416, 418, 421, 425, 463, 464,
\KV@glslink@hypertrue		478, 479, 482, 483, 493, 494,
	119, 134, 440, 450, 691, 701	497, 498, 502, 504, 620, 639,
		657, 658, 667, 669, 672, 675,
		676, 715, 730, 731, 734, 735,
		745, 746, 749, 750, 754, 756, 873
L		
\L	27, 384, 633	list (style)
\l	5, 27, 384, 633	274
\label	11, 225, 226,	\listcsadd
	228, 375, 500, 501, 625, 752, 753	220, 496, 748
\language	33, 388, 638	\listcsgadd
languages (option)	31	224, 225, 499, 751
\leaders	277, 578, 596, 831, 849	\listcsxadd
\leavevmode	94, 122, 422, 441, 673, 692	215, 216, 493, 745
\let	6, 7, 13,	listdotted (style)
	16–18, 20, 26, 27, 29–31, 33,	277
	35–43, 46, 47, 57, 69–71, 81,	\listead
	83, 94, 95, 97, 99, 100, 103–	220, 496, 748
	105, 107, 109, 112, 119, 120,	listgroup (style)
	122, 123, 125, 126, 133–140,	275
	156–163, 166, 174, 183, 184,	listhypergroup (style)
	188–191, 193, 195, 197, 200–	275
	203, 205, 209, 211–214, 223,	\listxadd
	225, 227, 244, 257, 258, 269,	84, 666
	270, 274, 288, 295, 296, 315,	\loadglsentries
	324, 326, 336, 337, 351–353,	112, 112, 433, 684
	372, 373, 376, 378–380, 383,	\long
	384, 386, 387, 389–394, 396,	94, 237, 422, 507, 673, 758
	400, 401, 408, 409, 414, 415,	long (key)
	422–426, 428–431, 433, 440–	80
	444, 449–453, 459–464, 467,	long (style)
	468, 473, 474, 477–483, 485–	278
	487, 489–491, 498, 499, 501,	long-booktabs (style)
	503, 509, 519–531, 538, 551,	284
	552, 565–570, 576, 587, 588,	long-sc-short (acrstyle)
	592, 593, 601, 605, 606, 617,	249
	622, 623, 626–629, 633–637,	long-sc-short-desc (acrstyle)
		250, 361
		long-short (acrstyle)
		247, 360
		long-short-desc (acrstyle)
		250, 361
		long-sm-short (acrstyle)
		249
		long-sm-short-desc (acrstyle)
		251, 362
		long-sp-short (acrstyle)
		247
		long-sp-short-desc (acrstyle)
		250
		long3col (style)
		279
		long3col-booktabs (style)
		285
		long3colborder (style)
		280
		long3colheader (style)
		281
		long3colheaderborder (style)
		281
		long4col (style)
		281
		long4col-booktabs (style)
		285
		long4colborder (style)
		282
		long4colheader (style)
		282

long4colheaderborder (style)	282	\MakeUppercase	
longaccess (key)	335		545, 546, 552, 798–800, 805, 806
longborder (style)	279	\marginpar	8, 374, 623
longheader (style)	279	\markboth	52, 397, 647
longheaderborder (style)	279	\mbox	176, 276, 299, 300, 321, 468, 578, 595, 608, 620, 720, 831, 849, 861, 862, 874
\longnewglossaryentry 94, 95, 422, 673		mcolalttree (style)	299
longplural (key)	80	mcolalttreegroup (style)	299
longpluralaccess (key)	335	mcolalttreehypergroup (style)	299
\longprovideglossaryentry	95, 422, 673	mcolalttreespannav (style)	300
longragged (style)	289	mcolindex (style)	295
longragged-booktabs (style)	286	mcolindexgroup (style)	295
longragged3col (style)	291	mcolindexhypergroup (style)	295
longragged3col-booktabs (style)	287	mcolindexspannav (style)	296
longragged3colborder (style)	292	mcoltree (style)	296
longragged3colheader (style)	292	mcoltreegroup (style)	297
longragged3colheaderborder (style)	292	mcoltreehypergroup (style)	297
longraggedborder (style)	290	mcoltreenoname (style)	297
longraggedheader (style)	290	mcoltreenonamegroup (style)	298
longraggedheaderborder (style)	291	mcoltreenonamehypergroup (style)	298
\LT@end@pen	288, 601, 855	mcoltreenonamespannav (style)	298
\LT@err	288, 601, 855	mcoltreespannav (style)	297
\LT@foot	288, 601, 602, 855	\memUChad	52, 397, 647
\LT@head	288, 601, 602, 855	\MessageBreak	23, 39, 71, 212, 324, 325, 333, 382, 390, 391, 409, 491, 535, 570, 587, 631, 640, 641, 659, 743, 789, 823, 824, 840
\LT@lastfoot	288, 601, 602, 855	mfirstuc (option)	41
\LT@output	288, 601, 855	\mfirstucMakeUppercase	
M			5, 372, 397, 420, 434–438, 453–463, 510, 515–517, 524, 528, 548–550, 553–556, 560, 561, 563, 591, 592, 622, 647, 648, 670, 685– 689, 705–713, 715, 761, 762, 767–769, 777, 781, 801–803, 806–809, 813, 814, 816, 844, 845
\makeatletter		\MFUaddmap	5
	85, 214, 416, 492, 667, 743	\MFUblocker	5
\makebox	277, 320–322, 578, 582, 583, 596, 620, 621, 831, 836, 849, 873, 874	\MFUexcl	5
\makefirstuc	4	\MFUsentencecase	4, 5, 41, 164, 165, 169
\makeglossaries	9, 34–36, 39, 40, 44, 78, 188, 192, 194–196, 198, 215, 374, 388–391, 394, 412, 477, 480– 482, 484, 492, 624, 638–641, 644, 662, 729, 732–734, 736, 744	\midrule	285, 286, 600, 853, 854
\makeglossary	38, 38, 39, 40, 390, 391, 640, 641	\month	179, 183, 470, 473, 572, 575, 722, 725, 826, 828
makeindex (option)	33	N	
\makenoidxglossaries	9, 34, 35, 78, 193, 194, 197, 198, 374, 388, 389, 412, 481– 484, 624, 638, 639, 662, 732–736	\n	183, 473, 575, 576, 725, 828, 829
\MakeTextUppercase	5, 372, 622	name (key)	75

<code>\NeedsTeXFormat</code>	4, 268, 271, 273, 277, 284, 289, 294, 300, 307, 314, 323–325, 332, 371, 372, 535, 570, 571, 576, 587, 622, 788, 823, 824, 830, 840, 848	. . . 19, 20, 40, 41, 72, 73, 193, 379, 380, 391, 392, 409, 410, 480, 629, 630, 642, 659, 660, 732
<code>\new@glossaryentry</code>	83, 84, 196, 415, 416, 482, 666, 734	<code>\newglossaryentry</code> 9, 41, 80, 83, 83, 107, 244, 374, 392, 413, 415, 430, 509, 519, 521, 522, 524, 526, 528, 530, 532, 565–568, 624, 642, 663–666, 681, 761, 771, 773, 775, 777, 779, 781, 783, 785, 818–821
<code>\new@ifnextchar</code>	72, 89, 90, 109–111, 134–138, 140–163, 242, 243, 328–332, 409, 419, 420, 432, 450–463, 507, 508, 589–591, 659, 670, 683, 702–714, 759, 760, 842–844	<code>\newglossarystyle</code> 234, 271, 274–287, 289–313, 315–320, 323, 505, 593, 595–621, 757, 846, 848–854, 856–873, 875
<code>\newacronym</code> 238, 239, 239, 244, 507, 509, 520, 522, 523, 525, 527, 529, 531, 533, 759, 761, 772, 774, 776, 778, 780, 782, 784, 786		<code>\newif</code> 6, 7, 22, 29, 32, 36, 107, 204, 372, 373, 381, 386, 388, 389, 487, 622, 630, 635, 638, 639, 739
<code>\newacronymhook</code> 244, 244, 369, 509, 520, 522, 524, 525, 527, 530, 531, 533, 565, 761, 772, 774, 776, 778, 780, 783, 784, 786, 818		<code>\newignoredglossary</code> . . . 74, 410, 660
<code>\newacronymstyle</code>	246, 247–252, 254–256, 510–516, 518, 762–768, 770	<code>\newlength</code> 133, 277, 278, 289, 301, 308, 318, 449, 596, 602, 609, 613, 618, 701, 849, 850, 856, 862, 863, 866, 867, 872
<code>\newcommand</code> 4, 5, 8–33, 35–37, 39–45, 47–90, 93–95, 100–107, 109–112, 115, 117–123, 125, 126, 133–178, 184, 187–192, 194, 197–203, 205–212, 214–220, 222–224, 226–235, 237–248, 257–259, 261–274, 288, 295, 314, 315, 320, 326–332, 334, 338–342, 359, 369–371, 373–383, 385–407, 409–422, 426–433, 436, 437, 439–444, 449–470, 474, 477–481, 483–490, 492–505, 507–512, 519–536, 540–542, 557, 565, 569, 570, 576, 588–595, 601, 605, 616, 617, 620, 623–633, 635–637, 639–644, 646–651, 653–657, 659–673, 677–684, 687–695, 701–721, 726, 729–733, 735–764, 771–789, 793–795, 810, 818, 822, 823, 830, 841–848, 855, 859, 870, 873		<code>\newrobustcmd</code> 8, 9, 67–69, 83, 84, 89–93, 109–111, 120, 121, 134–162, 164, 165, 169, 175, 176, 209, 210, 233, 236, 240–243, 260, 320, 328–331, 342–345, 374, 406–408, 415, 416, 419, 420, 432, 440, 450–469, 489, 490, 507, 508, 542–544, 588–591, 619, 623, 624, 656–658, 665, 666, 670, 683, 691, 702–720, 741, 759, 760, 795–797, 841–844, 873
<code>\newcount</code> . . 17, 81, 378, 414, 628, 664		<code>\newterm</code> 41, 392, 642
<code>\newcounter</code> 14, 15, 377, 626, 627		<code>\newtoks</code> 127, 188, 244, 444, 477, 509, 695, 729, 761
<code>\newenvironment</code> 229, 502, 754		<code>\newwrite</code> . . 35, 84, 179, 183, 188, 192, 389, 416, 470, 473, 477, 480, 639, 666, 722, 725, 729, 732
<code>\newglossary</code>		<code>\nfss@text</code> 8, 373, 623
		<code>\noalign</code> 288, 601, 855
		<code>\nobreak</code> 276, 288, 578, 595, 602, 831, 849, 855
		<code>\noexpand</code> 21, 43, 55, 56, 85, 99, 100, 123–126, 132, 133, 174, 185–187, 189–191, 193, 202, 207, 210, 214, 215, 217, 218, 237, 239, 244, 245, 380, 393, 399, 400,

581–583, 595, 605–609, 616– 621, 646, 755, 756, 831, 835, 836, 848, 849, 859–862, 870–875	
parent (key)	78
\parindent	295–300, 315, 317–319, 321–323, 581– 583, 605–609, 617–621, 834– 836, 859–862, 870–872, 874, 875
\parskip	295–298, 315, 317, 318, 605– 607, 617–619, 859–861, 870–872
\PassOptionsToPackage	325, 332, 535, 587, 788, 840
\pdfstringdefPreHook	203
\penalty	288, 601, 855
\PGLS	331, 591, 844
\Pgls	329, 329, 590, 843
\pgls	328, 329, 589, 842
\PGLSpl	331, 591, 844
\Pglspl	330, 330, 590, 843
\pglspl	329, 330, 589, 842
\phantomsection	53, 398, 648
plural (key)	76
pluralaccess (key)	334
\printacronyms	20, 380, 629
\PrintChanges	6, 372, 622
\printglossaries	194, 212, 481, 490, 733, 742
\printglossary	20, 24, 40, 41, 194, 211, 212, 227, 380, 382, 391, 392, 481, 490, 500, 630, 632, 642, 733, 742, 752
\printindex	41, 392, 642
\printnoidxglossaries	195, 212, 482, 490, 734, 742
\printnoidxglossary	194, 195, 198, 212, 212, 218, 219, 227, 481, 482, 484, 490, 495, 496, 500, 733, 734, 736, 742, 747, 748, 752
\printnumbers	40, 392, 642
\printsymbols	40, 391, 642
\ProcessOptions	325, 332, 535, 587, 788, 840
\ProcessOptionsX	42, 392, 642
\protect	54, 117, 165, 247–249, 254, 269, 350, 351, 360, 361, 366, 399, 439, 511, 512, 517, 521, 524, 526, 551, 557, 558, 562, 649, 689, 690, 763, 764, 769, 773, 777, 779, 804, 810, 811, 815
\protected@csedef	91, 420, 671
\protected@csxappto	270
\protected@csxdef	91, 270, 420, 671
\protected@eappto	35, 74, 75, 101, 202, 639, 660, 661, 678, 738
\protected@edef	8, 11, 18, 21, 35, 56, 58, 61, 63, 64, 67, 71, 72, 74, 75, 91–97, 100–102, 117, 122, 124, 165, 174–176, 195, 200, 204, 206, 207, 210, 215, 216, 226, 228, 234, 237, 244, 260, 269, 341, 342, 369, 374, 375, 400, 401, 403, 405, 424, 426, 427, 433–435, 439, 442, 443, 468, 485–487, 500, 502, 503, 505, 509, 528, 532, 536, 541, 542, 572, 573, 588, 592, 623, 625, 628, 630, 639, 650, 652, 654, 655, 657, 659–661, 671–675, 677, 678, 684–686, 690, 692–694, 719, 720, 733, 737–739, 742, 744, 745, 752– 755, 757, 761, 781, 785, 789, 794, 795, 818, 825, 826, 841, 845
\protected@write	71, 73, 179–181, 192, 193, 195, 198, 200, 210, 269, 409, 470, 471, 480, 481, 484, 485, 490, 573, 592, 659, 660, 722, 723, 732, 733, 736, 737, 742, 826, 845
\protected@xdef	16–18, 21, 26, 82, 97, 100, 102, 103, 206, 213, 214, 337, 338, 378–380, 383, 414, 427, 428, 487, 538– 540, 627, 628, 630, 633, 665, 675, 677–679, 739, 743, 792, 793
\providecommand	5, 20, 34, 44, 45, 52, 71, 109, 134, 181, 192, 195, 214, 215, 274, 295, 314, 328, 333, 380, 388, 394, 397, 409, 432, 450, 471, 480, 482, 484, 492, 502, 503, 536, 589, 594, 605, 616, 629, 638, 644, 648, 659, 682, 702, 723, 732, 734, 736, 744, 754, 755, 789, 842, 848, 859, 870
\provideglossaryentry	83, 416, 666
\ProvidesFile	45, 394, 644
\ProvidesGlossariesLang	45, 394, 644
\ProvidesPackage	4, 268, 271,

273, 278, 284, 289, 295, 300, 308, 314, 324, 325, 332, 371, 372, 535, 570, 571, 576, 587, 592–594, 596, 600, 602, 605, 609, 613, 616, 622, 788, 823, 824, 830, 840, 845, 846, 848, 850, 853, 856, 859, 862, 866, 870	
R	
<code>\r</code> 26, 383, 633	<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, 107, 108, 174, 176–178, 184, 185, 192–197, 213, 215, 225, 239, 244, 245, 247–256, 271, 272, 275–283, 285, 286, 288–305, 308–313, 315–323, 333, 337, 346, 348, 350, 351, 358–369, 372–377, 379–382, 385–392, 394–396, 404, 410, 412, 422, 430, 431, 468–470, 474, 475, 480–483, 491, 492, 500, 509–520, 522, 523, 525–527, 529–531, 533, 536, 538, 544, 547, 549–551, 556–568, 571, 572, 576–586, 593–611, 613–615, 617–627, 629–632, 634–638, 640, 641, 643, 645, 646, 654, 661–663, 673, 681, 682, 719–721, 726, 727, 732–735, 743, 744, 751, 752, 761–776, 778, 780, 782– 784, 786, 789, 791, 792, 797, 800, 802, 804, 809–821, 824, 825, 829–839, 846–865, 867–875
<code>\raggedright</code> 287, 289–294, 308–314, 601– 605, 613–616, 854–859, 867–870	<code>\renewenvironment</code> 229, 271, 275, 278–284, 286, 287, 289–315, 317, 318, 321, 502, 593, 595–599, 601– 620, 754, 846, 848, 850–872, 874
<code>\raisebox</code> 133, 449, 701	<code>\renewglossarystyle</code> 235, 505, 757
<code>\ref</code> 228, 501, 753	<code>\RequireGlossariesLang</code> 45, 46, 324, 325, 394, 395, 570, 587, 644, 645, 823, 840
<code>\refstepcounter</code> 228, 501, 753	<code>\RequirePackage</code> 4– 6, 12, 13, 29, 30, 42, 45, 261, 278, 284, 289, 295, 300, 308, 324, 325, 333, 372, 376, 386, 392, 394, 533, 535, 536, 570, 587, 596, 600, 602, 605, 609, 613, 622, 625, 626, 635, 636, 642, 644, 786, 788, 789, 823, 840, 850, 853, 856, 859, 862, 866
<code>\regex</code> 233	<code>\restorecounters@</code> 124, 442, 693
<code>\relax</code> 7, 10, 13, 17, 18, 20, 30, 31, 35, 36, 39, 41, 42, 57, 71, 76, 78, 82, 85, 97, 100, 104, 108, 109, 120, 124, 125, 127–132, 166, 181, 183–185, 187, 188, 191, 193, 195, 197, 201, 206, 207, 209, 212, 214, 222, 225, 233, 234, 262, 270, 274, 288, 295, 299, 300, 314, 316, 317, 319– 323, 336, 337, 351–353, 373, 375, 376, 378–380, 386, 387, 389, 391–393, 401, 409, 411, 412, 415, 416, 424, 426, 428, 431, 440–449, 464, 472–477, 480–483, 486–489, 491, 497, 499, 500, 504, 505, 533, 538, 551, 552, 565–568, 572, 574– 576, 581–583, 592, 595, 601, 605, 608, 609, 616–622, 625, 626, 628, 629, 636, 639–643, 651, 659, 661, 662, 665, 667, 675, 677, 679, 681, 682, 691, 693–700, 715, 716, 724–726, 728, 729, 731–735, 738–741, 743, 749, 751, 756, 757, 786, 791, 792, 804, 805, 818–820, 822, 825, 827–829, 834–836, 845, 848, 855, 859, 862, 870–875	<code>\restoremakegloss</code> (option) 34
<code>\renewacronymstyle</code> 246, 360– 363, 365–368, 510, 557–560, 562–564, 762, 810–813, 815, 817	<code>\romannumeral</code> 202, 205, 206, 320, 322, 486, 487, 583, 620, 621, 738, 739, 836, 873, 874

S	
\s@glshyp@opt . . .	120, 120, 440, 691
\s@GlsSetXdyFirstLetterAfterDigits	177, 469, 721
\s@GlsSetXdyNumberGroupOrder	178, 469, 470, 721
\s@ifglossaryexists	64, 64, 211, 405, 490, 655, 656, 742
\s@newglossary	72, 72, 409, 659
sanitize (option)	29
sanitizesort (option)	28
\savecounters@	124, 442, 693
savenumberlist (option)	12
savewrites (option)	38
sc-short-long (acrstyle)	249
sc-short-long-desc (acrstyle)	251, 362
section (option)	10
see (key)	77
seeautonumberlist (option)	12
\seename	45, 209, 394, 489, 644, 741
seencoidex (option)	9
\setabbreviationstyle	261, 784
\SetAcronymLists	22, 381, 630
\SetAcronymStyle	261, 387, 388, 531, 532, 637, 784, 785
\setacronymstyle	239, 246, 261, 510, 762, 784
\setbool	28, 385, 635
\setbox	288, 601, 602, 855
\setcounter	228, 501, 753
\SetCustomDisplayStyle	261, 532, 533, 785, 786
\SetCustomStyle	261, 533, 786
\SetDefaultAcronymDisplayStyle	258, 519, 520, 771, 772
\SetDefaultAcronymStyle	259, 520, 531, 772, 784
\SetDescriptionAcronymDisplayStyle	260, 524, 525, 776, 778
\SetDescriptionAcronymStyle	260, 525, 531, 778, 785
\SetDescriptionDUAAcronymDisplayStyle	259, 522–524, 775, 776
\SetDescriptionDUAAcronymStyle	259, 523, 531, 776, 785
\SetDescriptionFootnoteAcronymDisplayStyle	259, 520, 522, 773–775
\SetDescriptionFootnoteAcronymStyle	259, 522, 531, 774, 785
\SetDUADisplayStyle	261, 530, 531, 783, 784
\SetDUASyle	261, 531, 532, 784, 785
\setentrycounter	56, 181, 224, 234, 400, 471, 499, 504, 571, 650, 723, 751, 756, 824
\SetFootnoteAcronymDisplayStyle	260, 526, 527, 778, 780
\SetFootnoteAcronymStyle	260, 527, 532, 780, 785
\SetGenericNewAcronym	244, 246, 509, 510, 761, 762
\setglossarypreamble	50, 397, 647
\setglossarysection	52, 398, 648
\setglossarystyle	212, 225, 234, 262, 275–277, 279–287, 290–300, 302–307, 309–311, 313, 316, 318, 319, 323, 491, 504, 505, 533, 595–601, 603–619, 621, 743, 756, 757, 786, 848–873, 875
\setglossentrycompatibility	231, 499, 503, 505, 751, 755, 757
\setkeys	29, 33, 42, 52, 96, 122, 175, 176, 213, 244, 385, 388, 393, 398, 423, 441, 468, 469, 491, 509, 520, 522, 523, 525, 527, 529, 531, 533, 635, 638, 643, 648, 674, 692, 720, 743, 761, 772, 774, 776, 778, 780, 782, 784, 786
\setlength	277, 278, 289, 295–298, 301, 308, 315, 317, 318, 322, 583, 596, 602, 605–607, 609, 613, 617–621, 836, 849, 850, 856, 859–863, 866, 867, 870–872, 874
\SetSmallAcronymDisplayStyle	260, 528–530, 781–783
\SetSmallAcronymStyle	261, 529, 532, 782, 785
\setstylefile	48, 396, 646
\settoheight	133, 449, 701
\settowidth	248, 320–322, 512, 582, 583, 620, 621, 764, 836, 873, 874
\setupglossaries	42, 392, 643
stylecode	13, 376, 626
short (key)	80
short-long (acrstyle)	248, 360
short-long-desc (acrstyle)	251, 362

shortaccess (key)	<u>335</u>	\showglotype	<u>263</u> , <u>533</u> , <u>787</u>
shortcuts (option)	<u>32</u>	\showglouser	<u>263</u> , <u>534</u> , <u>787</u>
shortplural (key)	<u>80</u>	\showglouserii	<u>264</u> , <u>534</u> , <u>787</u>
shortpluralaccess (key)	<u>335</u>	\showglouseriii	<u>264</u> , <u>534</u> , <u>787</u>
\show	<u>262–267</u> , <u>370</u> , <u>371</u> , <u>533–</u> <u>535</u> , <u>569</u> , <u>570</u> , <u>786–788</u> , <u>822</u> , <u>823</u>	\showglouseriv	<u>264</u> , <u>534</u> , <u>787</u>
\showacronymlists	<u>266</u> , <u>535</u> , <u>788</u>	\showglouserv	<u>264</u> , <u>534</u> , <u>787</u>
\showglocounter	<u>263</u> , <u>533</u> , <u>787</u>	\showglouservi	<u>264</u> , <u>534</u> , <u>787</u>
\showglodesc	<u>265</u> , <u>534</u> , <u>787</u>	sm-short-long (acrstyle)	<u>249</u>
\showglodescaccess	<u>370</u> , <u>570</u> , <u>823</u>	sm-short-long-desc (acrstyle)	<u>251</u> , <u>363</u>
\showglodescplural	<u>265</u> , <u>534</u> , <u>787</u>	smallcaps (option)	<u>32</u>
\showglodescpluralaccess		smaller (option)	<u>32</u>
.		\SmallNewAcronymDef	<u>260</u> , <u>369</u> , <u>528</u> , <u>530</u> , <u>568</u> , <u>781</u> , <u>783</u> , <u>821</u>
.	<u>370</u> , <u>570</u> , <u>823</u>	sort (key)	<u>76</u>
\showglofield	<u>266</u> , <u>535</u> , <u>788</u>	sort (option)	<u>15</u>
\showglofirst	<u>263</u> , <u>533</u> , <u>787</u>	\space	<u>9</u> , <u>34–36</u> , <u>39</u> , <u>44</u> , <u>55</u> , <u>56</u> , <u>60</u> , <u>63</u> , <u>71</u> , <u>78</u> , <u>80</u> , <u>107</u> , <u>110</u> , <u>111</u> , <u>117</u> , <u>173</u> , <u>179–183</u> , <u>187</u> , <u>189–195</u> , <u>197</u> , <u>198</u> , <u>202</u> , <u>210</u> , <u>212</u> , <u>215</u> , <u>229</u> , <u>241</u> , <u>247–253</u> , <u>255</u> , <u>256</u> , <u>260</u> , <u>270</u> , <u>273</u> , <u>275</u> , <u>276</u> , <u>278</u> , <u>290</u> , <u>301</u> , <u>308</u> , <u>309</u> , <u>315–317</u> , <u>319</u> , <u>321</u> , <u>322</u> , <u>350</u> , <u>351</u> , <u>360–369</u> , <u>374</u> , <u>388–391</u> , <u>394</u> , <u>400</u> , <u>403</u> , <u>405</u> , <u>412</u> , <u>413</u> , <u>430</u> , <u>432</u> , <u>433</u> , <u>439</u> , <u>440</u> , <u>467</u> , <u>470–473</u> , <u>477–484</u> , <u>489</u> , <u>491</u> , <u>492</u> , <u>501</u> , <u>502</u> , <u>505</u> , <u>508</u> , <u>511–</u> <u>519</u> , <u>524</u> , <u>528</u> , <u>550</u> , <u>551</u> , <u>557–</u> <u>565</u> , <u>571–575</u> , <u>577–585</u> , <u>593–</u> <u>597</u> , <u>602</u> , <u>609</u> , <u>613</u> , <u>617–621</u> , <u>624</u> , <u>638–641</u> , <u>644</u> , <u>650</u> , <u>653</u> , <u>655</u> , <u>662–664</u> , <u>681</u> , <u>683</u> , <u>689–</u> <u>691</u> , <u>719</u> , <u>722–725</u> , <u>729–736</u> , <u>741</u> , <u>743</u> , <u>744</u> , <u>753</u> , <u>757</u> , <u>759</u> , <u>763–771</u> , <u>777</u> , <u>781</u> , <u>804</u> , <u>810–</u> <u>818</u> , <u>824–828</u> , <u>831</u> , <u>833–839</u> , <u>846–850</u> , <u>856</u> , <u>863</u> , <u>867</u> , <u>870–874</u>
\showglofirstaccess	<u>370</u> , <u>569</u> , <u>822</u>	\spacefactor	<u>13</u> , <u>376</u> , <u>626</u>
\showglofirsttpl	<u>263</u> , <u>533</u> , <u>787</u>	\SS	<u>27</u> , <u>384</u> , <u>634</u>
\showglofirstpluralaccess		\ss	<u>27</u> , <u>384</u> , <u>634</u>
.	<u>370</u> , <u>569</u> , <u>823</u>	\str	<u>233</u>
\showgloflag	<u>266</u> , <u>534</u> , <u>788</u>	\string	<u>9</u> , <u>18</u> , <u>19</u> , <u>24</u> , <u>27</u> , <u>34–36</u> , <u>39</u> , <u>44</u> , <u>54–56</u> , <u>58–61</u> , <u>63</u> , <u>71</u> , <u>73</u> , <u>78</u> , <u>80</u> , <u>86</u> , <u>89</u> , <u>90</u> , <u>101</u> , <u>102</u> , <u>107</u> , <u>109–111</u> , <u>124</u> , <u>125</u> , <u>127–129</u> , <u>132</u> , <u>173</u> , <u>177–184</u> , <u>187</u> , <u>188</u> , <u>192–198</u> , <u>202</u> , <u>207</u> , <u>208</u> , <u>212</u> , <u>214</u> , <u>215</u> , <u>218</u> , <u>219</u> , <u>227</u> , <u>239</u> ,
\showgloindex	<u>266</u> , <u>534</u> , <u>788</u>		
\showglolevel	<u>262</u> , <u>533</u> , <u>786</u>		
\showgloloclist	<u>266</u> , <u>535</u> , <u>788</u>		
\showglolong	<u>266</u> , <u>534</u> , <u>788</u>		
\showglolongaccess	<u>371</u> , <u>570</u> , <u>823</u>		
\showglolongpluralaccess			
.	<u>371</u> , <u>570</u> , <u>823</u>		
\showglongame	<u>264</u> , <u>534</u> , <u>787</u>		
\showglongameaccess	<u>370</u> , <u>569</u> , <u>822</u>		
\showgloparent	<u>262</u> , <u>533</u> , <u>786</u>		
\showgloplural	<u>263</u> , <u>533</u> , <u>786</u>		
\showglopluralaccess	<u>370</u> , <u>569</u> , <u>822</u>		
\showgloshort	<u>265</u> , <u>534</u> , <u>787</u>		
\showgloshortaccess	<u>371</u> , <u>570</u> , <u>823</u>		
\showgloshortpluralaccess			
.	<u>371</u> , <u>570</u> , <u>823</u>		
\showglosort	<u>265</u> , <u>534</u> , <u>787</u>		

261, 269, 371, 372, 374, 379, 382, 388–391, 394, 399–405, 409, 412, 413, 416, 417, 419, 427, 430, 432, 433, 439, 440, 443, 445, 446, 448, 467, 469– 474, 476, 477, 480–484, 488, 489, 491, 492, 495, 496, 500, 503–505, 571–576, 592, 624, 629, 632, 638–641, 644, 649, 650, 652–655, 659, 660, 662– 664, 667–670, 678, 681–683, 690, 691, 694, 696–698, 700, 719–726, 728, 729, 732–736, 740, 741, 743, 744, 747, 748, 752, 755, 757, 784, 824–829, 845	
\strut	275– 278, 280, 282, 290, 291, 293, 301, 303, 305, 309, 310, 312, 319, 504, 577–580, 582, 584– 586, 595–598, 602–604, 609– 611, 613–615, 619, 756, 831– 833, 835, 837–839, 848–852, 856–858, 863–865, 867–869, 873
style (option)	11
styles:	
altlist	276
altlistgroup	276
altlisthypergroup	276
altlong4col	283
altlong4col-booktabs	286
altlong4colborder	283
altlong4colheader	283
altlong4colheaderborder	284
altlongragged4col	292
altlongragged4col-booktabs	287
altlongragged4colborder	294
altlongragged4colheader	293
altlongragged4colheaderborder	294
altsuper4col	306
altsuper4colborder	307
altsuper4colheader	306
altsuper4colheaderborder	307
altsuperragged4col	312
altsuperragged4colborder	313
altsuperragged4colheader	313
altsuperragged4colheaderborder	313
alttree	320
alttreegroup	323
alttreehypergroup	323
index	315
indexgroup	316
indexhypergroup	316
inline	271
list	274
listdotted	277
listgroup	275
listhypergroup	275
long	278
long-booktabs	284
long3col	279
long3col-booktabs	285
long3colborder	280
long3colheader	281
long3colheaderborder	281
long4col	281
long4col-booktabs	285
long4colborder	282
long4colheader	282
long4colheaderborder	282
longborder	279
longheader	279
longheaderborder	279
longragged	289
longragged-booktabs	286
longragged3col	291
longragged3col-booktabs	287
longragged3colborder	292
longragged3colheader	292
longragged3colheaderborder	292
longraggedborder	290
longraggedheader	290
longraggedheaderborder	291
mcolalttree	299
mcolalttreegroup	299
mcolalttreehypergroup	299
mcolalttreespannav	300
mcolindex	295
mcolindexgroup	295
mcolindexhypergroup	295
mcolindexspannav	296
mcoltree	296
mcoltreegroup	297
mcoltreehypergroup	297
mcoltreenoname	297
mcoltreenonamegroup	298
mcoltreenonamehypergroup	298
mcoltreenonamespannav	298
mcoltreespannav	297

<code>\textsmaller</code>	249, 250, 256, 368, 513, 518, 522, 526, 527, 530, 564, 765, 770, 775, 778, 780, 783, 817	449, 474–476, 490, 502, 503, 506, 536, 571, 644, 645, 650, 654, 655, 659, 695–701, 726– 728, 742, 754, 755, 758, 789, 824
<code>\texttt</code>	238, 507, 759	
<code>\textulc</code>	240, 507, 759	<code>\toprule</code> 285, 286, 600, 853, 854
<code>\textup</code>	238, 240, 507, 759	<code>\TrackLanguageTag</code> 45
<code>\TH</code>	27, 384, 634	<code>\trans@languages</code> 46, 394, 644
<code>\th</code>	27, 384, 634	<code>\translate</code> 46, 47, 395, 645, 646
<code>\the</code>	43, 46, 56, 61, 63, 72, 127–132, 179, 183, 185, 187, 199, 200, 205, 210, 222, 237, 244, 245, 247, 248, 253, 254, 360, 364, 366, 369, 393, 395, 400, 403, 405, 409, 444–449, 470, 473, 475, 476, 484, 485, 487, 490, 497, 502, 503, 506, 509–512, 516, 517, 519–530, 532, 536, 557, 558, 561, 562, 565–569, 571, 572, 575, 644, 645, 650, 654, 655, 659, 695–701, 722, 725– 728, 736, 737, 739, 742, 749, 754, 755, 758, 761–764, 767, 769, 771–775, 777, 779, 781– 783, 785, 786, 789, 810, 811, 814, 815, 818–822, 824, 826, 828	<code>translate (option)</code> 30 <code>\translatelet</code> 19, 20, 40, 379, 380, 391, 392, 629, 630, 642
<code>\the@numberlist</code>	174, 468, 719, 720	<code>tree (style)</code> 317
<code>\theglossary</code>	6, 372, 622	<code>treegroup (style)</code> 318
<code>\theglossary (env.)</code>	229	<code>treehypergroup (style)</code> 318
<code>\theglossaryentry</code>	14, 229, 377, 501, 626, 753	<code>treenoname (style)</code> 318
<code>\theglossarysubentry</code>	15, 229, 377, 502, 627, 753	<code>treenonamegroup (style)</code> 319
<code>\theglentrycounter</code>	123, 124, 203, 206, 442, 486, 487, 572, 693, 738, 739, 825	<code>treenonamehypergroup (style)</code> 319
<code>\theH</code>	208, 488, 740	<code>\triangleright</code> 8, 374, 623
<code>\theHglossaryentry</code>	14, 377, 626	<code>\ttfamily</code> 8, 374, 623
<code>\theHglossarysubentry</code>	15, 377, 627	<code>\TX@trial</code> 103, 428, 679
<code>\theHglentrycounter</code>	124, 203, 206, 442, 443, 486, 487, 693, 738, 739	<code>type (key)</code> 77
<code>\thesection</code>	43, 393, 643	<code>\typeout</code> 23, 381, 631
<code>\this@dialect</code>	46, 47, 324, 325, 395, 570, 587, 645, 823, 840	
<code>\tiny</code>	8, 374, 623	
<code>\tl</code>	5	
<code>toc (option)</code>	10	
<code>\toks@</code>	43, 46, 56, 61, 63, 72, 127– 132, 185, 187, 210, 237, 393– 395, 400, 403, 405, 409, 444–	

U

<code>\u</code>	26, 383, 633
<code>\uccode</code>	222, 497, 749
<code>ucmark (option)</code>	14
<code>\undef</code>	79, 85, 211, 412, 416, 490, 663, 667, 742
<code>\unexpanded</code>	41
<code>\unskip</code>	94, 277, 422, 578, 596, 673, 831, 849
<code>\unvbox</code>	288, 601, 602, 855
<code>\usedictionary</code>	46, 394, 645
<code>\usepackage</code>	27, 218, 219, 239, 371, 372, 495, 496, 747, 748
<code>user1 (key)</code>	79
<code>user1access (key)</code>	335
<code>user2 (key)</code>	79
<code>user2access (key)</code>	335
<code>user3 (key)</code>	79
<code>user3access (key)</code>	335
<code>user4 (key)</code>	79
<code>user4access (key)</code>	336
<code>user5 (key)</code>	79
<code>user5access (key)</code>	336
<code>user6 (key)</code>	80
<code>user6access (key)</code>	336
<code>\UTFviii@two@octets</code>	27, 384, 634

<code>\UTFviii@two@octtets@combine</code> . . .	X
27, 384, 634	<code>\x</code> 506
V	<code>\xatlevel@</code> 123, 442, 693
<code>\v</code> 26, 384, 633	<code>\xcapitalisewords</code> 465
<code>\vbox</code> 288, 601, 602, 855	<code>\xdef</code> 98,
<code>\vsize</code> 288, 602, 855	424, 426, 491, 592, 675, 845, 846
<code>\vskip</code> . . . 274, 288, 295, 314, 595,	<code>\xglsaccsupp</code> . . 342, <u>342</u> , 541, 542, 795
601, 605, 616, 848, 855, 859, 870	<code>\xglsfieldaccsupp</code> <u>341</u> , 541, 794
<code>\vss</code> 288, 601, 602, 855	<code>\xifinlist</code> 84, 666
W	<code>\xifinlistcs</code> 215–
<code>\warn@nomakeglossaries</code>	217, 220, 493, 496, 744, 745, 748
. <u>191</u> , 193–195, 480–482, 731–734	<code>xindy</code> (option) <u>33</u>
<code>\warn@noprintglossary</code>	<code>xindygloss</code> (option) <u>34</u>
. . . . 193–195, <u>210</u> , 214, 480–	<code>xindynoglsnumbers</code> (option) <u>34</u>
482, 490, 491, 732–734, 742, 743	<code>\xmakefirstuc</code>
<code>\write</code> . 35, 86, 109, 179–184, 189– 433–435, 439, 463, 464,
192, 195, 199, 214, 215, 389,	588, 684–686, 690, 715, 716, 841
416, 417, 432, 470–474, 478,	<code>\xspace</code> 239, 507, 759
479, 482, 485, 492, 572–576,	
639, 667, 682, 683, 722–726,	Y
730, 731, 734, 736, 744, 826–829	<code>\year</code> 179, 183, 470,
<code>writeglslabelnames</code> (option) <u>36</u>	473, 572, 575, 722, 725, 826, 828
<code>writeglslabels</code> (option) <u>36</u>	
<code>\writeist</code> <u>178</u> , 188, 470,	Z
473, 474, 477, 572, 575, 576,	<code>\Z</code> 233
722, 724, 726, 729, 826, 828, 829	<code>\z@</code> 288, 601, 602, 855