

Documented Code For glossaries v4.57

Nicola L.C. Talbot

Dickimaw Books

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

2025-03-12

Contents

1	Introduction	3
2	Main Package Code (glossaries.sty)	4
2.1	Package Definition	4
2.2	Package Options	7
2.3	Predefined Text	47
2.4	Xindy	58
2.5	Loops and conditionals	67
2.6	Defining new glossaries	74
2.7	Defining new entries	79
2.8	Resetting and unsetting entry flags	107
2.9	Keeping Track of How Many Times an Entry Has Been Unset . .	112
2.10	Loading files containing glossary entries	117
2.11	Using glossary entries in the text	117
	2.11.1 Links to glossary entries	123
	2.11.2 Displaying entry details without adding information to the glossary	169
2.12	Adding an entry to the glossary without generating text	181
2.13	Creating associated files	182
2.14	Writing information to associated files	204
2.15	Glossary Entry Cross-References	214
2.16	Displaying the glossary	216
2.17	Acronyms	254
2.18	Predefined acronym styles	260
2.19	Predefined Glossary Styles	278
2.20	Debugging Commands	279
2.21	Compatibility with version 2.07 and below	284

3	Glossary Styles	285
3.1	Glossary hyper-navigation definitions (<code>glossary-hypernav.sty</code>) . . .	285
3.2	In-line Style (<code>glossary-inline.sty</code>)	289
3.3	List Style (<code>glossary-list.sty</code>)	291
3.4	Glossary Styles using <code>longtable</code> (<code>glossary-long.sty</code>)	295
3.5	Glossary Styles using <code>longtable</code> and <code>booktabs</code> (<code>glossary-longbooktabs.sty</code>)	302
3.6	Glossary Styles using <code>longtable</code> (<code>glossary-longragged.sty</code>)	306
3.7	Glossary Styles using <code>multicol</code> (<code>glossary-mcols.sty</code>)	312
3.8	Glossary Styles using <code>supertabular</code> environment (<code>glossary-super.sty</code>)	318
3.9	Glossary Styles using <code>supertabular</code> environment (<code>glossary-superragged.sty</code>)	325
3.10	Tree Styles (<code>glossary-tree.sty</code>)	331
4	Multi-Lingual Support	341
4.1	Babel Captions (<code>glossaries-babel.sty</code>)	341
4.2	Polyglossia Captions (<code>glossaries-polyglossia.sty</code>)	342
5	Prefix Support (<code>glossaries-prefix</code> Code)	343
6	Accessibility Support (<code>glossaries-accsupp</code> Code)	350
6.1	Defining Replacement Text	351
6.2	Accessing Replacement Text	356
6.3	Displaying the Glossary	376
6.4	Acronyms	377
6.5	Debugging Commands	387
7	Backwards Compatibility	389
7.1	<code>glossaries-compatible-207</code>	389
7.2	<code>glossaries-compatible-307</code>	389
8	Rollback	390
8.1	Rollback v4.46 (<code>glossaries-2020-03-19.sty</code>)	390
8.2	Rollback v4.46 (<code>glossaries-accsupp-2020-03-19.sty</code>)	553
8.3	Rollback v4.46 (<code>glossaries-babel-2020-03-19.sty</code>)	588
8.4	Rollback v4.46 (<code>glossaries-compatible-207-2020-03-19.sty</code>)	588
8.5	Rollback v4.46 (<code>glossaries-compatible-307-2020-03-19.sty</code>)	594
8.6	Rollback v4.46 (<code>glossaries-polyglossia-2020-03-19.sty</code>)	604
8.7	Rollback v4.46 (<code>glossaries-prefix-2020-03-19.sty</code>)	605
8.8	Rollback v4.46 (<code>glossary-hypernav-2020-03-19.sty</code>)	609
8.9	Rollback v4.46 (<code>glossary-inline-2020-03-19.sty</code>)	611
8.10	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	612
8.11	Rollback v4.46 (<code>glossary-list-2020-03-19.sty</code>)	614
8.12	Rollback v4.46 (<code>glossary-longbooktabs-2020-03-19.sty</code>)	617
8.13	Rollback v4.46 (<code>glossary-longragged-2020-03-19.sty</code>)	620
8.14	Rollback v4.46 (<code>glossary-mcols-2020-03-19.sty</code>)	623
8.15	Rollback v4.46 (<code>glossary-super-2020-03-19.sty</code>)	626
8.16	Rollback v4.46 (<code>glossary-superragged-2020-03-19.sty</code>)	630

8.17	Rollback v4.46 (glossary-tree-2020-03-19.sty)	634
8.18	Rollback v4.49 (glossaries-2021-11-01.sty)	639
8.19	Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	806
8.20	Rollback v4.49 (glossaries-babel-2021-11-01.sty)	841
8.21	Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	841
8.22	Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	847
8.23	Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	857
8.24	Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	858
8.25	Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	862
8.26	Rollback v4.49 (glossary-inline-2021-11-01.sty)	864
8.27	Rollback v4.49 (glossary-list-2021-11-01.sty)	865
8.28	Rollback v4.49 (glossary-long-2021-11-01.sty)	867
8.29	Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	871
8.30	Rollback v4.49 (glossary-longragged-2021-11-01.sty)	873
8.31	Rollback v4.49 (glossary-mcols-2021-11-01.sty)	876
8.32	Rollback v4.49 (glossary-super-2021-11-01.sty)	880
8.33	Rollback v4.49 (glossary-superragged-2021-11-01.sty)	884
8.34	Rollback v4.49 (glossary-tree-2021-11-01.sty)	887
8.35	Rollback v4.52 (glossaries-2022-11-03.sty)	893
8.36	Rollback v4.52 (glossary-hypernav-2022-11-03.sty)	1046
9	Rollback	1047
9.1	Rollback v4.54 (glossaries-2024-04-03.sty)	1047

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` v4.57: L^AT_EX₂ ϵ Package to Assist Generating Glossaries”.

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

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

`INSTALL` Installation instructions.

`CHANGES` Change log.

README.md Package summary.

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

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

2 Main Package Code (glossaries.sty)

2.1 Package Definition

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossaries-2022-11-03.sty}
\DeclareRelease{v4.54}{2024-04-03}{glossaries-2024-04-03.sty}
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossaries}[2025/03/12 v4.57 (NLCT)]
```

Required packages:

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

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

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

```
\ExplSyntaxOn
```

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

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

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

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

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

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

Automatically identify exclusions, blockers and mappings.

```
\glsmfuexcl
```

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

```
\glsmfublocker
```

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

```
\glsmfuaddmap
```

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

```
}
{
```

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

```
\RequirePackage{textcase}
```

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

```
\MFUsentencecase
```

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

Provide support for exclusions with `\MFUsentencecase`.

```
\glsmfuexcl
```

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

```
\mfirstucMakeUppercase
```

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

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

Just treat blockers and mappings as exclusions.

```
\glsmfublocker
```

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

```

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

```

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

```
\RequirePackage{amsgen}
```

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

```
\RequirePackage{etoolbox}
```

Check if `doc` has been loaded.

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

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

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

```
{%
```

```
\@gls@docloadedtrue
```

```
}%
```

```
{%
```

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

```
}
```

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

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

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

```
\glsorg@theglossary
```

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

```
\glsorg@endtheglossary
```

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

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

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

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

```
\begingroup
```

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

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

```
\glsorg@PrintChanges
```

```
\endgroup
```

```
}
```

End of doc stuff.

\fi

2.2 Package Options

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

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

```

    \fi
  }

```

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

```

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

```

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

```

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

```

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

```

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

```

`\glsshowtargetouter`

```

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

```

`\glsshowtargetsymbol`

```

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

```

`\glsshowtargetfont`

```

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

```

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

```

\newcommand*\@glsshowtarget}[1]{

```

`\glsdoshowtarget` Make it easier to customise.

```

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

```

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

```

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

```

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

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

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

```
\newrobustcmd*\glsshowaccsupp}[3]{%
  \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}%
}
```

`seenoinde`

```
\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
}

```

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

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

```

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

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

```

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

```

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

```

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

```

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

```

Determine whether or not to use numbered sections.

\@@glossarysecstar

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

```

\@@glossaryseclabel

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

```

\glsautoprefix Prefix to add before label if automatically generated:

```
\newcommand*\glsautoprefix{}

```

numberedsection

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

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

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

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

`\glo@seeautonumberlist`

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

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

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

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

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

`\gls@loadlong`

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

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

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

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

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

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

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

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

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

`\@gls@loadlist`

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

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

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

`\@gls@loadtree`

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

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

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

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

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

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

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

`nopostdot` Boolean option to suppress post description dot

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

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

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

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

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

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

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

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

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

`\define@glossarysubentrycounter`

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

Check if counter already defined.

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

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

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

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

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

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

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

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

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

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

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

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

`\glsprestandardsort`

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

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

`\glo@check@sortallowed`

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

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

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

Store entry information when it's defined.

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

No count register required for standard sort.

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

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

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

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

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

```
\fi
```

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

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

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

```
}%
```

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

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

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

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

```
}
```

Set standard sort as the default:

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

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

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

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

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

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

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

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

```
\number#1%
```

```
}
```

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

```

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

```

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

```

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

```

Get the parent, if one exists

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

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

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

Set index information for this entry

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

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

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

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

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

Don't store entry index information.

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

No count register required for standard sort.

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

Don't modify sort value.

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

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

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

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

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

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

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

Don't store entry index information.

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

No count register required for standard sort.

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

Clear sort value.

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

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

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

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

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

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

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

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

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

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

`\glsdefaulttype`

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

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

`\acronymtype`

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

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

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

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

```
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%  
  \ifglsacronym  
    \renewcommand{\@gls@do@acronymsdef}{%  
      \DeclareAcronymList{acronym}%  
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%  
      \renewcommand*\acronymtype{acronym}%  
    }  
  Define hook to set the toc title when translator is in use.  
    \newcommand*\gls@tr@set@acronym@toctitle}{%  
      \translatelet{\glossarytoctitle}{Acronyms}%  
    }%  
  }%  
  \else  
    \let\@gls@do@acronymsdef\relax  
  \fi  
}
```

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

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

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

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

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

```
\@gls@declareoption{acronyms}{%  
  \glsacronymtrue  
  
  \def\@gls@do@acronymsdef{%  
    \DeclareAcronymList{acronym}%  
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%  
    \renewcommand*\acronymtype{acronym}%  
  }  
  Define hook to set the toc title when translator is in use.  
    \newcommand*\gls@tr@set@acronym@toctitle}{%  
      \translatelet{\glossarytoctitle}{Acronyms}%  
    }  
}
```

```

    }%
  }%
}

```

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

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

`\@addtoacronymlists`

```

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

```

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

```

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

```

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

`\glsIfListOfAcronyms`

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

```

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

```

Internal command requires label and list to be expanded:

```

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

```

Not found

```

  #4%
  \else

```

Found

```

  #3%
  \fi
}

```

```

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

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

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

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

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

\glscounter
    \newcommand{\glscounter}{page}

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

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

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

```

nohypertypes

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

`\GlossariesWarning` Prints a warning message.

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

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

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

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

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

`nowarn` Define package option to suppress warnings

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

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

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

`nolangwarn` Suppress warning if language support not found.

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

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

```
\@gls@warnontheGLOSSdefined Issue a warning if overriding theGLOSSary
  \newcommand*\@gls@warnontheGLOSSdefined}{%
    \GlossariesWarning{Overriding 'theGLOSSary' environment}%
  }
```

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

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

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

`\glssetexpandfield`

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

Sets field to always expand.

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

`\glssetnoexpandfield`

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

Sets field to never expand.

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

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

```
\glssetexpandfield{type}
```

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

```
\glssetnoexpandfield{desc}
```

```

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

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

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

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

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

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

    Sanitizing stuff:

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

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

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

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

```

`\@gls@noidx@nosanitizesort` Check for the existence of `\datatool_sort_preprocess:Nn`

```

\ExplSyntaxOn
\cs_if_exist:NTF \datatool_sort_preprocess:Nn
{
  \newcommand*{\@gls@noidx@nosanitizesort}
  {
    \ifdefvoid\@glo@sort
    { }
    {
      \cs_if_exist:NTF \datatool_sort_preprocess:NnN
      {
        \exp_args:NNV
        \datatool_sort_preprocess:NnN
        \@glo@sort
        \@glo@sort
        \c_false_bool
      }
      {
        \exp_args:NNV
        \datatool_sort_preprocess:Nn
        \@glo@sort
        \@glo@sort
      }
    }
  }
}

```

Old method.

```

\newcommand*{\@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@nosanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@nosanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
}
\ExplSyntaxOff

```

`\glsnoidxstripaccents` This is now deprecated due to changes in the L^AT_EX kernel.

This strips accents by redefining the standard accent commands to just do their argument. (This will be localised since `\glsnoidxstripaccents` is used within a group.) Anything outside this standard set really shouldn't be using

\makenoidxglossaries. It's much better to use xindy or bib2gls with the correct language setting.

```

\newcommand*{\glsnoidxstripaccents}{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let~\@firstofone
  \let"\@firstofone
  \letu\@firstofone
  \lett\@firstofone
  \letd\@firstofone
  \letr\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone
  \letv\@firstofone
  \letH\@firstofone
  \letc\@firstofone
  \letb\@firstofone

  \leta\@secondoftwo
  \def\AE{AE}%
  \def\ae{ae}%
  \def\OE{OE}%
  \def\oe{oe}%
  \def\AA{AA}%
  \def\aa{aa}%
  \def\L{L}%
  \def\l{l}%
  \def\O{O}%
  \def\o{o}%
  \def\SS{SS}%
  \def\ss{ss}%
  \def\th{th}%

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

```

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

```
\ifl@t@r\fmtversion{2019/10/01}
```

```
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
```

`\@gls@removedoption`

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

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

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

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

\define@boolkey[gls]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifgls@sanitize@symbol
    \glssetnoexpandfield{symbol}%
    \glssetnoexpandfield{symbolplural}%
  \else
    \glssetexpandfield{symbol}%
    \glssetexpandfield{symbolplural}%
  \fi
}
```

`sanitizesort`

```
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
  \ifglssanitizesort
    \glssetnoexpandfield{sortvalue}%
  }
```

```

\renewcommand*\@gls@noidx@setsanitizesort}{%
  \glssanitizesorttrue
  \glsssetnoexpandfield{sortvalue}%
}%
\else
  \glsssetexpandfield{sortvalue}%
  \renewcommand*\@gls@noidx@setsanitizesort}{%
    \glssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
  }%
\fi
}

```

Default setting:

```

\glssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%

```

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

```

\newcommand*\@gls@noidx@setsanitizesort}{%
  \glssanitizesortfalse
  \glsssetexpandfield{sortvalue}%
}

\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
  \setbool{glssanitizesort}{#1}%
  \ifglssanitizesort
    \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}%
      \glsssetexpandfield{name}%
      \glsssetexpandfield{symbol}%
      \glsssetexpandfield{symbolplural}%
      \glsssetexpandfield{desc}%
      \glsssetexpandfield{descplural}%
    }%
  }%
  \setkeys[gls]{sanitize}{#1}%
}%
}

```

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

```
\newif\ifglstranslate
```

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

```
\@gls@usetranslator
```

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

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

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

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

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

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

```
\define@choicekey{glossaries.sty}{translate}%
[\gls@translate@val\gls@translate@nr]%
{true,false,babel}[true]%
{%
```

```

\ifcase\gls@translate@nr\relax
\glstranslatetrue
\renewcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
}
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}

```

Set the default value:

```

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

```

`\@gls@requested@languages`

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

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

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

locales Provide synonym.

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

indexonlyfirst Set whether to only index on first use.

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

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

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

shortcuts Define acronym shortcuts.

```

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

```

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

```

\newcommand*\glsorder}{word}

```

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

```

\newcommand*\@glsorder}{1}{}

```

order

```

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

```

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

```

\newif\ifglsxindy

```

The default is `makeindex`:

```

\glsxindyfalse

```

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

```

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

```

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

```
\define@boolkey[glS]{xindy}{glsnumbers}[true]{%
\glSxindy@glSnumberstrue
```

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

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

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

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

Define key to set the language

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

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

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

Define a key to set the code page.

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

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

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

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

```
\@glS@declareoption{xindygloss}{%
\glSxindytrue
}
```

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

```

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

```

`\@domakeglossaries`

```

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

```

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

```

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

```

`restoremakegloss` Cancel the effect of `disablemakegloss`.

```

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

```

```

    }%
  }%
}

```

`\@do@write@glslabels`

```

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

```

`\@@do@write@glslabels`

```

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

```

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

```

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

```

`\@write@glslabelsname`

```

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

```

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

```

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

```

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

```

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

\ifglsautomake
\newif\ifglsautomake

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

```

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

```

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

```

`\gls@automake@option`

```

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

```

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

```

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

```

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

```

\define@choicekey{glossaries.sty}{automake}{%
[\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\gls@automake@nr\relax
\GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
case 0: true (same as delayed)
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
case 1: false
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
case 2: immediate
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or

```

```

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

```

`\@gls@doautomake`

```

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

```

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

```

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

```

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

```

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

```

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

```
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \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}{\glsymbolsgroupname}%
\newcommand*{\printsymbols}[1][\]{\printglossary[type=symbols,##1]}%

```

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

```

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

```

numbers Create a “symbols” glossary type

```

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

```

Default is not to define the numbers glossary:

```

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

```

\@gls@numbersdef

```

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

```

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

```

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

```

index Create an “index” glossary type

```

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

```

```

\fi
}

```

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

```

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

```

Default is not to define index glossary:

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Now process options passed to the package:

```
\ProcessOptionsX
```

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

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

```
\@gls@runshell
```

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

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

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

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

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

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

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

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

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

```
\@gls@run@output@file
```

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

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

```
\ifglsautomake
```

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

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

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

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

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

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

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

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

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

```

{
  #1 ~ \glsquote { \l_glossaries_output_dir_tl }
}

```

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

```

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

```

Delayed write

```

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

```

Immediate write

```

\cs_set:Npn \@gls@runshell #1 { \sys_shell_now:e { #1 } }
\cs_set_eq:NN \@gls@loadshellesc \relax
}
}
{
  \cs_set_eq:NN \@gls@loadshellesc \relax
  \legacy_if_set_false:n { glsautomake }
  \PackageWarning { glossaries }
  {
    Can't ~ implement ~ automake: ~ no ~ shell
  }
}
\ExplSyntaxOff
\@gls@loadshellesc
\fi

```

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

```

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

```

Now define `\setupglossaries`:

```

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

```

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}`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\RequireGlossariesLang`

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

`\ProvidesGlossariesLang`

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

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

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

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

```
\ifglstranslate
```

Load tracklang:

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

Track any languages explicitly requested via the languages option.

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

Load translator if required.

```
\@gls@usetranslator
```

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

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

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

```
\ifboolexpr
{
  test {\ifdefstring{\trans@languages}{English}}
  and not
  test {\ifdefstring{\bbl@loaded}{english}}
}
{%
```

```

\let\glsifusetranslator\@secondoftwo
}%
{%
\usedictionary{glossaries-dictionary}%
\renewcommand*{\addglossarytocaptions}[1]{%
\ifcsundef{captions#1}{%
{%
\expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
\expandafter\toks@\expandafter{\@gls@tmp
\renewcommand*{\glossaryname}{\translate{Glossary}}}%
}%
\expandafter\edef\csname captions#1\endcsname{\the\toks@}%
}%
}%
}%
}%
{}%

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

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

```

```

        \translate{Description (glossaries)}}%
\renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
\renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
\renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
\renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
}{}%
\fi

```

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

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

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

```
\DeclareRobustCommand*\nopostdesc{}

```

`\@nopostdesc` Suppress next description terminator.

```

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

```

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

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

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

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

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

```

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

```

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

```
\@onlypremakeg\setStyleFile
```

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

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

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

The `makeglossaries` Perl script picks up this name from the auxiliary file. If the name ends with `.xdy` it calls `xindy` otherwise it calls `makeindex`. Since its not required by 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 loca-
tion numbers are in the form <letter><compositor><number>. For example, if
\@glsAlphacompositor is set to "." then it allows locations such as A.1 whereas
if \@glsAlphacompositor is set to "-" then it allows locations such as A-1.
  \newcommand*\@glsAlphacompositor}{\glscompositor}
```

`\glsSetAlphaCompositor` Sets the alpha compositor.

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

Can only be used before `\makeglossaries`

```
\@onlypremakeg\glsSetAlphaCompositor
```

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

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

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

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

Only has an effect when used before `\makeglossaries`

```
\@onlypremakeg\glsSetSuffixF
```

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

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

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

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

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

```
\ifcsundef{hyperlink}%
{%
  \newcommand*\{glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*\{glsnumberformat}[1]{\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]{\cuse{#2}{\cuse{#1}{#3}}}
```

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

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

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

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

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

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

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

```
\setglossarypreamble
```

Code provided by Michael Pock.

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

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

```

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

`\glossarypostamble`

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

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

```

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

  \glsglossarymark{\glossarytoctitle}%
}
```

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

```

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

```

    }
  }
}

```

`\glossarymark` Provided for backward compatibility:

```

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

```

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

`\setglossarysection`

```

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

```

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

`\@glossarysection`

```

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

```

Do automatic labelling if required

```

  \@@glossaryseclabel
}

```

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

`\@p@glossarysection`

```

\newcommand*\@p@glossarysection[2]{%
  \glsclearpage
  \phantomsection

```

```

\ifdefempty\@glossarysecstar
{%
  \csname\@glossarysec\endcsname{#2}%
}%
{%
  \@gls@toc{#1}{\@glossarysec}%
  \csname\@glossarysec\endcsname*{#2}%
}%
Do automatic labelling if required
\@glossaryseclabel
}

```

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

```

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

```

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

```

\newcommand*\glscclearpage{\gls@doclearpage}

```

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

```

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

```

2.4 Xindy

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

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

\ifglsxindy
  \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.

```

\ifglxindy

```

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

```

\newcommand*\@glsaddxdyattribute[2]{%

```

Add to xindy attribute list

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

Add to xindy markup location.

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

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

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

High-level command:

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

Add to comma-separated attribute list

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

First add the attribute without the counter name:

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

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

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

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

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

Only has an effect before `\writeist`:

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

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

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

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

`\@xdyuseralphabets` List of additional alphabets

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

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

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

This code is only required for xindy:

```
\ifglxindy
```

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

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

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

    End of xindy-only block:
    \fi

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

\GlsAddXdyLocation \GlsAddXdyLocation[prefix-loc]{name}{definition} Define a new loca-
tion called name. The definition must use xindy syntax. (Note that this
doesn't check to see if the location is already defined. That is left to xindy to
complain about.)
    \ifglxindy
    \newcommand*\GlsAddXdyLocation}[3] [] {%
        \def\@gls@tmp{#1}%
        \ifx\@gls@tmp\@empty
        \edef\@xdyuserlocationdefs{%
            \@xdyuserlocationdefs ^^J%
            (define-location-class \string"#2\string"^^J\space\space
            \space(:sep \string"{}\@glsopenbrace\string" #3
            :sep \string"\@glsclosebrace\string"))
        }
    }

```

```

    }%
\else
  \edef\@xdyuserlocationdefs{%
    \@xdyuserlocationdefs ^^J%
    (define-location-class \string"#2\string"^^J\space\space
      \space(:sep "\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]{%
    \glsnoinindywarning\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]{%
    \glsnoinindywarning\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]{%
        \glsnoindywarning\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]{%
        \glsnoindywarning\GlsAddXdyStyle}
    \fi

\GlsSetXdyStyles Reset the list of required styles
  \ifglxindy
    \newcommand*\GlsSetXdyStyles[1]{%
      \edef\@xdyrequiredstyles{#1}}
  \else
    \newcommand*\GlsSetXdyStyles[1]{%
      \glsnoindywarning\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 com-
mand for makeglossaries to pick up the information from the auxiliary file.
This command is not needed by the glossaries package, so define it to ignore its
arguments.
  \def\@xdylanguage#1#2{}

```

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

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

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

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

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

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

Suggested by egreg:

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

`\@xdylettergroups` Store letter group definitions.

```
\ifglxindy
  \ifglxindy@glsnumbers
    \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}%
}

```

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

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

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

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\glsdetoklabel`

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

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

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

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

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

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

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

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

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

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

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

`\glsdoifexists`

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

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

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

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

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

`\glsdoifexistsorwarn`

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

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

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

`\glsdoifexistsordo`

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

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

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

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

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

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

`\ifglshaschildren`

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

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

```

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

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

```

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

\ifglshasparent

```

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

```

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

```

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

```

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

```

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

```

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

```

\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@glstdetoklabel{#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@glstdetoklabel{#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@glstdetoklabel{#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{\ifgl$fieldvoid}[4]{\ifcvoid{glo@glstdetoklabel{#2}@#1}{#3}{#4}}

```

```

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

```

\ifgl\$hasfield

```

\newrobustcmd*{\ifgl$hasfield}[4]{%

```

```

\glsdoifexists{#2}%
{%
  \letcs{\@glo@thisvalue}{glo@glstoklabel{#2}@#1}%

```

First check supplied field label is defined.

```

  \ifdef\@glo@thisvalue
  {%

```

Is defined, so now check if empty.

```

    \ifdefempty\@glo@thisvalue
    {%

```

Is empty, so doesn't have field set.

```

      #4%
    }%
  }%

```

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

```

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

```

Value is set to the default value.

```

      #4%
    }%
  }%

```

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

```

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

```

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

```

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

```

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

```

  \ifdef\@gls@thisfield
  {%

```

Is defined, so now check if empty.

```

    \letcs{\@glo@thisvalue}{glo@glstoklabel{#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@defaultttype`

```
\newcommand*{\@gls@defaultttype}{\glsdefaultttype}
```

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

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

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

```
\GlossariesWarning{\string\defglsentryfmt\space attempting to set
a display style for \string\glsdefaulttype\space but there
doesn't seem to be a default glossary. If you have used
'nomain', you need to define an alternative glossary}%
\else
\csgdef{gls@\@glo@tmp @entryfmt}{#2}%
\fi
}
```

`\gls@doentryfmt`

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

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

```
\newcommand*{\@gls@forbidtext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}
```

`\gls@gobbleopt` Discard optional argument.

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

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

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

where *<log-ext>* is the extension of the `makeindex` transcript file, *<in-ext>* is the extension of the glossary input file (read in by `\printglossary` and created by

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

`\newglossary`

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

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

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

`\ns@newglossary` Define the unstarred version.

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

Check if default has been set

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

Add this to the list of glossary types:

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

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

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

Store the file extensions:

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

Store the title:

```
  \expandafter\def\csname @glo@type@#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`

```
\@onlypremakegnewglossary
```

`\@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 @glsotype@#1@counter\endcsname{#2}%
}

```

Add counter to xindy list, if not already added:

```

\ifglxindy
  \GlsAddXdyCounters{#2}%
\fi
}

```

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

`\@gls@getcounter`

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

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

```
\glsdefmain
```

Define the “acronym” glossaries if required.

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

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

```
\@gls@do@symbolsdef
```

```
\@gls@do@numbersdef
```

```
\@gls@do@indexdef
```

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

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

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

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

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

```

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

```

2.7 Defining new entries

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

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

```

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

```

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

```

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

```

descriptionplural

```

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

```

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

```

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

```

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

```

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

```

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

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

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

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

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

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

`\gls@default@value`

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

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

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

`symbolplural`

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

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

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

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

```
\define@key{glossentry}{counter}{%
```

```

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

```

see The see key specifies a list of cross-references

```

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

```

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

`\gls@set@xr@key`

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

```

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

```

`\gls@xr@key`

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

`\gls@checkseeallowed`

```

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

```

`\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\@gls@detoklabel{##1}@nonumberlist{\@glo@nonumberlist}%
  }%
  {}%
}%
\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}

```

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

user1

```

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

```

user2

```

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

```

user3

```

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

```

user4

```

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

```

user5

```

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

```

user6

```

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

```

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

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

```

        \fi
    }

\glsdefaultsort Define command to set default sort.
    \newcommand*\@glsdefaultsort{\@glo@name}

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

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

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

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

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

@gls@expand@fields
    \newcommand{\@gls@expand@fields}[4]{%

```

```

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

```

\@gls@startswithexpandonce

```

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

```

```

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

```

\gls@assign@field

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

```

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

```

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

```

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

```

`\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` $\langle label \rangle$ $\langle key-val list \rangle$. There are two required fields in $\langle key-val list \rangle$: 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.

```
\ifglsentryexists{#1}%
{%
  \xifinlist{#1}{\@gls@docdef@labellist}%
  {%
```

This entry has already been defined in the document environment.

```
\GlossariesWarning{Entry '#1' has already been defined in the
  document environment (ignoring new definition)}%
}%
{%
  \xifinlist{#1}{\@gls@glsdefs@labellist}%
  {%
```

This entry was defined in the `.glsdefs` file so update the definition in case it has been changed.

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

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

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

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

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

```

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

```

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

Write key value information:

```

\@for@gls@map:=@gls@keymap\do
{%

```

```

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

```

Provide hook:

```

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

```

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

```

\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}

```

`\@gls@fetchfield`

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

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

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

Ensure user field name is fully expanded

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

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

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

Found it.

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

Break out of loop.

```
\@endfortrue  
}%  
{}%  
}%  
}
```

`\glsaddstoragekey`

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

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

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

Starred version switches on expansion for this key.

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

Unstarred version doesn't override default expansion.

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

Check the specified key doesn't already exist.

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

Set up the key.

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

Set the default value.

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

Assignment code.

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

Define the no-link commands.

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

```
\glsaddkey{<key>}{<default value>}{<no link cs>}{<no link  
ucfirst  
cs>}{<link cs>}{<link ucfirst cs>}{<link allcaps cs>}
```

\glsaddkey

Allow user to add their own custom keys.

```
\newcommand*{\glsaddkey}{\ifstar\@sglsaddkey\@glsaddkey}
```

Starred version switches on expansion for this key.

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

Unstarred version doesn't override default expansion.

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

Check the specified key doesn't already exist.

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

Set up the key.

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

Set the default value.

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

Assignment code.

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

Define the no-link commands.

```
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%  
\newcommand*{#4}[1]{\@Gls@entry@field{##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}{#3{##2}##3}%  
  }%  
  \newrobustcmd*{#5}{%  
    \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%  
}%
```

```

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

\glsfielddedef

```

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

\glsfieldgdef

```

\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}{}%
    }%
  }%
}

```

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

\glsfieldddef

```

\newrobustcmd{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%

```

```

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

```

`\glsfieldfetch`

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

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

```

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

```

`\glsunexpandedfieldvalue`

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

For use in expandable contexts where the field value is required, but the contents should not be expanded. The `<field>` argument must be the internal field name.

```

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

```

`\ifglsfieldeq`

`\ifglsfieldeq{<label>}{<field>}{<string>}{<true>}{<false>}`

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

```

\newrobustcmd{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%

```

```

\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{
  \ifcsstring{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
{
  \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}

```

```

\ifglsfielddefeq{<label>}{<field>}{<command>}{<true>}
{<false>}

```

\ifglsfielddefeq

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

```

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

```

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

```

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

```

```

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

```

\ifglsfieldcseq

As above but uses \ifcsstrequal instead of \ifdefstrequal

```

\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{
  \protected@edef\glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\glo@label @#2}%
  {%
    \ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
  }%
  {%

```

```

        \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
}

```

`\glswritedefhook`

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

```

`\gls@assign@desc`

```

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

```

`\longnewglossaryentry`

```

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

```

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

```
\@onlypreamble{\longnewglossaryentry}

```

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

```

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

```

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

```

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

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

```

Prevent any further use of `\GlsSetQuote`:

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

Store label

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

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

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

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

```
\let\@glo@name\@gls@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\glsnodesc
  \let\@glo@desc\@empty
\fi
}%
{}%
}%
\protected@edef\glolist@\@glo@type{\csname glolist@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
  \@glolist@\@glo@label},}%
}%

```

Initialise level to 0.

```
\gls@level=0\relax
```

Has this entry been assigned a parent?

```
\ifx\@glo@parent\@empty
```

Doesn't have a parent. Set `\glo@<label>@parent` to empty.

```
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%  
\else
```

Has a parent. Check to ensure this entry isn't its own parent.

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

Check the parent exists:

```
\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\@gls@name  
  \expandafter\let\expandafter\@glo@name  
  \csname glo@\@glo@parent @name\endcsname
```

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

```
\ifx\@glo@plural\@gls@default@value  
  \expandafter\let\expandafter\@glo@plural  
  \csname glo@\@glo@parent @plural\endcsname  
\fi  
\fi  
}%  
{%
```

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

```
\PackageError{glossaries}%  
{%  
  Invalid parent ‘\@glo@parent’  
  for entry ‘\@glo@label’ - parent doesn't exist%  
}%  
{%
```

```

        Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
\fi

```

Set the level for this entry

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

Define commands associated with this entry:

```

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

```

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

```

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

```

```

\ifcsundef{@glo@type@\@glo@type @counter}%
  {%
    \def\@glo@defaultcounter{\glscounter}%
  }%
  {%
    \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
  }%
\gls@assign@field{\@glo@defaultcounter}\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{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\@glsn@name
  \@glsn@name
  \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 nonnumberlist 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\@glsn@desc
  \@glsn@desc
  \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{\glstoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
  \def\glo@list{,%}
  \forlgsentries[\glo@type]{\glo@label}%
  {%
    \ifdequal\glo@thislabel\glo@label
      {}{\protected@eappto\glo@list{\glo@label,}}%
    }%
  \cslet\glo@list@\glo@type{\glo@list}%
  \csdef\glo@\glo@thislabel @type{#2}%
}
```

`\@glossaryentryfield` Indicate what command should be used to display each entry in the glossary. (This enables the glossaries-accsupp package to use `\accsuppglossaryentryfield` instead.)

```
\ifglxindy
  \newcommand*\@glossaryentryfield{\string\glossentry}
\else
  \newcommand*\@glossaryentryfield{\string\glossentry}
\fi
```

`\@glossarysubentryfield` Indicate what command should be used to display each subentry in the glossary. (This enables the glossaries-accsupp package to use `\accsuppglossarysubentryfield` instead.)

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

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

`\@glo@storeentry`

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

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

Escape `makeindex/xindy` special characters in the label:

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

Get the sort string and escape any special characters

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

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

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

Get the parent, if one exists

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

Write the information to the glossary file.

```
\ifglsxindy
```

Store using xindy syntax.

```
\ifx\@glo@parent\@empty
```

Entry doesn't have a parent

```
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%  
(\string\@glo@sort\string" %  
\string\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %  
}%  
\else
```

Entry has a parent

```
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%  
\csname glo@\@glo@parent @index\endcsname  
(\string\@glo@sort\string" %  
\string\@glo@prefix\@glossarysubentryfield  
{\csname glo@#1@level\endcsname}\@glo@esclabel}\string") %  
}%  
\fi  
\else
```

Store using makeindex syntax.

```
\ifx\@glo@parent\@empty
```

Sanitize \@glo@prefix

```
\@onelevel@sanitize\@glo@prefix
```

Entry doesn't have a parent

```
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%  
\@glo@sort\@gls@actualchar\@glo@prefix  
\@glossaryentryfield{\@glo@esclabel}%  
}%  
\else
```

Entry has a parent

```
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%  
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar  
\@glo@sort\@gls@actualchar\@glo@prefix  
\@glossarysubentryfield  
{\csname glo@#1@level\endcsname}\@glo@esclabel}%  
}%
```

```

    \fi
  \fi
}

```

2.8 Resetting and unsetting entry flags

Each glossary entry is assigned a conditional of the form `\ifglo@<label>@flag` which determines whether or not the entry has been used (see also `\ifglsused` defined below). These flags can be set and unset using the following macros, but first we need to know if we're in `amsmath`'s `align` environment's measuring pass.

```

\gls@if@measuring
  \newcommand*{\@gls@if@measuring}[2]{%
    \ifmeasuring@
      #1%
    \else
      #2%
    \fi
  }

```

```

\gls@org@refstepcounter
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label

```

```

\gls@if@measuring
  \newcommand{\gls@if@measuring}[2]{#2}
  If amsmath has been loaded, set \gls@if@measuring to \@gls@if@measuring
  \AtBeginDocument{%
    \@ifpackageloaded{amsmath}%
      {\let\gls@if@measuring\@gls@if@measuring}%
      {}%
    \let\gls@org@refstepcounter\refstepcounter
    \let\gls@org@label\label
  }

```

```

\gls@ifnotmeasuring
  \newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring{#1}}

```

```

\glsifmeasuring
  \newcommand*\glsifmeasuring{\gls@if@measuring}

```

```

\gls@local@stepcounter
  \newcommand*{\gls@local@stepcounter}[1]{%
    \ifcsdef{c@#1}%
      {%
        \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
        \expandafter\advance\csname c@#1\endcsname by \@ne
      }
    }

```

```

}%
{}%
}

```

`\gls@start@measuring` It's best to scope this, but still provide some support to switch on and off.

```

\newcommand{\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\@glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo
  \let\@glslink\@secondoftwo
}

```

If datatool-base v3.0 available, add to measuring hook:

```

\ExplSyntaxOn
\tl_if_exist:NT \l_datatool_measure_hook_tl
{
  \tl_put_right:Nn \l_datatool_measure_hook_tl
  { \gls@start@measuring }
}
\ExplSyntaxOff

```

`\gls@stop@measuring`

```

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

```

`\glsmeasureheight`

```

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

```

`\glsmeasuredepth`

```

\newcommand{\glsmeasuredepth}[2]{%
  \settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}

```

`\glsmeasurewidth`

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

`\glspatchtabularx` Patch `\TX@trial` (as per David Carlisle's answer in <http://tex.stackexchange.com/a/94895>). This does nothing if `\TX@trial` hasn't been defined.

```
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\glspatchtabularx\relax
  }%
  {}%
}
```

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

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

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

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

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

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

```

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

```

`\glslocalunset` As above, but with only a local effect:

```

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

```

`\@glslocalunset` Local unset. This defaults to just `\@@glslocalunset` but is changed by `\glsenableentrycount`.

```

\newcommand*\@glslocalunset{\@@glslocalunset}

```

`\@@glslocalunset` Local unset without checks.

```

\newcommand*\@@glslocalunset[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}

```

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

```

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

```

`\@@glsunset` Global unset without checks.

```

\newcommand*\@@glsunset[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}

```

`\glslocalreset` Local reset. This defaults to just `\@@glslocalreset` but is changed by `\glsenableentrycount`.

```

\newcommand*\glslocalreset{\@@glslocalreset}

```

`\@@glslocalreset` Local reset without checks.

```

\newcommand*\@@glslocalreset[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}

```

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

```

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

```

`\@@glsreset` Global reset without checks.

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

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

`\glsresetall`

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

As above, but with only a local effect:

`\glslocalresetall`

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

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

`\glsunsetall`

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

As above, but with only a local effect:

`\glslocalunsetall`

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

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

Version 4.14 introduced `\glsenableentrycount` that keeps track of how many times an entry is marked as used. The counter is reset back to zero when the first use flag is reset. Note that although the word “counter” is used here, it’s not an actual 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 `\gls`. (Only global unset is used if previous count was one as it doesn't make sense to have a local unset here given that the previous count was global.)

```

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

```

Similarly for the analogous commands. No case change plural:

```

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

```

First letter uppercase singular:

```

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

```

First letter uppercase plural:

```

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

```

Write information to aux file at the end of the document

```
\AtEndDocument{\@gls@write@entrycounts}%
```

Fetch previous count information from aux file. (No check here to determine if the entry is still defined.)

```

\renewcommand*{\@gls@entry@count}[2]{%
  \csdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%

```

`\glsenableentrycount` may only be used once and only in the preamble.

```

\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount

```

`\@gls@increment@currcount`

```

\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
  }

```

`s@local@increment@currcount`

```

\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
  }

```

`\@gls@write@entrycounts` Write the entry counts to the aux file. Use `\immediate` since this occurs right at the end of the document. Only write information for entries that have been used. (Some users have a file containing vast numbers of entries, many of which may not be used. There's no point writing information about the entries that haven't been used and it will only slow things down.)

```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
    }%
  }%
}

```

`\@gls@entry@count` Default behaviour is to ignore arguments. Activated by `\glsenableentrycount`.

```
\newcommand*{\@gls@entry@count}[2]{}
```

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

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

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

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

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

```
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
  \string\gls\space since you haven't enabled entry counting}%
  \@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*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
```

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

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

Add sentence case mapping.

```
\glsmfuaddmap{\cgl}{\cGls}
```

`\@cG`ls 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}[]}%
}
```

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

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

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

```

\newcommand*\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Add sentence case mapping.

```

\glsmfuaddmap{\cglsp1}{\cGlspl}

```

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

```

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

```

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

```

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

```

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

```

\newcommand*\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}

```

2.10 Loading files containing glossary entries

Glossary entries can be defined in an external file. These external files can contain `\newglossaryentry` and `\newacronym` commands.¹

```
\loadglsentries[<type>]{<filename>}
```

This command will input the file using `\input`. The optional argument specifies to which glossary the entries should be assigned if they haven't used the `type` key. If the optional argument is not specified, the default glossary is used. Only those entries used in the document (via `\glslink`, `\gls`, `\glspl` and uppercase variants or `\glsadd` and `\glsaddall` will appear in the glossary). The mandatory argument is the filename (with or without `.tex` extension).

```

\loadglsentries
\newcommand*\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}

```

`\loadglsentries` can only be used in the preamble:

```
\@onlypreamble{\loadglsentries}
```

2.11 Using glossary entries in the text

Any term that has been defined using `\newglossaryentry` (or `\newacronym`) can be displayed in the text (i.e. outside of the glossary) using one of the commands defined in this section. Unless you use `\glslink`, the way the term appears in the text is determined by `\glsdisplayfirst` (if it is the first time the term has been used) or `\glsdisplay` (for subsequent use). Any formatting commands (such as `\textbf` is governed by `\glsformat`. By default this just displays the *link text* “as is”.

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

```

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

\glsentryfmt As from version 3.11a, the way in which an entry is displayed is now governed by
\glsentryfmt. This doesn't take any arguments. The required information is
set by commands like \gls. This used to use the now deprecated \glsdisplay
and \glsdisplayfirst. Replaced with \glsgenentryfmt in v4.50.
    \newcommand*{\glsentryfmt}{\glsgenentryfmt}

\glsgenentryfmt Define a generic format that just uses the first, text, plural or first plural keys
(or the custom text) with the insert text appended.
    \newcommand*{\glsgenentryfmt}{%
        \ifdefempty\glscustomtext
            {%
                \glsifplural
                    {%
                        Plural form
                            \glscapscase
                                {%
                                    Don't adjust case
                                        \ifglsused\glslabel
                                            {%
                                                Subsequent use
                                                    \glsentryplural{\glslabel}\glsinsert
                                                        }%
                                                        {%
                                                            First use
                                                                \glsentryfirstplural{\glslabel}\glsinsert
                                                                    }%
                                                                    }%
                                                                    {%
                                                                        Make first letter upper case
                                                                            \ifglsused\glslabel
                                                                                {%
                                                                                    Subsequent use.
                                                                                        \Glsentryplural{\glslabel}\glsinsert
                                                                                            }%
                                                                                            {%
                                                                                                First use
                                                                                                    \Glsentryfirstplural{\glslabel}\glsinsert
                                                                                                        }%
                                                                                                        }%
                                                                                                        {%

```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glsuppercase{\glsentryplural{\glslabel}\glsinsert}%  
}%  
{%
```

First use

```
\glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%  
}%  
}%  
{%
```

Singular form

```
\glscapscase  
{%
```

Don't adjust case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glsentrytext{\glslabel}\glsinsert  
}%  
{%
```

First use

```
\glsentryfirst{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\Glsentrytext{\glslabel}\glsinsert  
}%  
{%
```

First use

```
\Glsentryfirst{\glslabel}\glsinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glsuppercase{\glsentrytext{\glslabel}\glsinsert}%  
}%  
{%
```

First use

```
\glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%  
}%  
}%  
}%  
}%  
{%
```

Custom text provided in `\glsdisp`. (The insert is most likely to be empty at this point.)

```
\glscustomtext\glsinsert  
}%  
}
```

`\glsacronymfont` Define a generic acronym format that uses the long and short keys (or their plurals) and `\acrffullformat`, `\firstacronymfont` and `\acronymfont`.

```
\newcommand*{\glsacronymfont}{%  
  \ifdefempty\glsacronymfonttext  
  {%  
    \ifglsused\glslabel  
    {%
```

Subsequent use:

```
\glsifplural  
}%
```

Subsequent plural form:

```
\glsifplural  
}%
```

Subsequent plural form, don't adjust case:

```
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert  
}%  
}%
```

Subsequent plural form, make first letter upper case:

```
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert  
}%  
}%
```

Subsequent plural form, all caps:

```
\glsuppercase  
  {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%  
}%  
}%  
}%
```

Subsequent singular form

```
\glscapscase  
{%
```

Subsequent singular form, don't adjust case:

```
\acronymfont{\glsentryshort{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent singular form, make first letter upper case:

```
\acronymfont{\Glsentryshort{\glslabel}}\glsinsert  
}%  
{%
```

Subsequent singular form, all caps:

```
\glsuppercase  
  {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%  
}%  
}%  
{%
```

First use:

```
\glsifplural  
{%
```

First use plural form:

```
\glscapscase  
{%
```

First use plural form, don't adjust case:

```
\genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, make first letter upper case:

```
\Genplacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use plural form, all caps:

```
\glsuppercase  
  {\genplacrfullformat{\glslabel}{\glsinsert}}}%  
}%  
}%  
{%
```

First use singular form

```
\glscapscase  
{%
```

First use singular form, don't adjust case:

```
\genacrfullformat{\glslabel}{\glsinsert}%
```

```
}%  
{%
```

First use singular form, make first letter upper case:

```
\Genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, all caps:

```
\glsuppercase  
{\genacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
}%  
}%  
{%
```

User supplied text.

```
\glscustomtext  
}%  
}
```

```
\genacrfullformat{<label>}{<insert>}
```

\genacrfullformat

The full format used by \glsacrfmt (singular).

```
\newcommand*{\genacrfullformat}[2]{%  
  \glsentrylong{#1}#2\space  
  (\protect\firstacronymfont{\glsentryshort{#1}})%  
}
```

```
\Genacrfullformat{<label>}{<insert>}
```

\Genacrfullformat

As above but makes the first letter upper case.

```
\newcommand*{\Genacrfullformat}[2]{%  
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%  
  \expandafter\glsentencecase\expandafter{\gls@text}%  
}
```

```
\genplacrfullformat{<label>}{<insert>}
```

\genplacrfullformat

The full format used by \glsplacrfmt (plural).

```
\newcommand*{\genplacrfullformat}[2]{%  
  \glsentrylongpl{#1}#2\space  
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%  
}
```

```
\Genplacrfullformat{<label>}{<insert>}
```

`\Genplacrfullformat`

As above but makes the first letter upper case.

```
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
```

`\glsdisplayfirst` `\glsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\glsdisplay` `\glsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplay` `\defglsdisplay` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

`\defglsdisplayfirst` `\defglsdisplayfirst` deprecated in v3.11a (2013-10-15) and removed in v4.50. Use rollback if required.

2.11.1 Links to glossary entries

The links to glossary entries all have a first optional argument that can be used to change the format and counter of the associated entry number. Except for `\glslink` and `\glsdisp`, the commands like `\gls` have a final optional argument that can be used to insert additional text in the link (this will usually be appended, but can be redefined using `\defglsentryfmt`). It goes against the L^AT_EX norm to have an optional argument after the mandatory arguments, but it makes more sense to write, say, `\gls{label}[’s]` rather than, say, `\gls[append=’s]{label}`. Since these control sequences are defined to include the final square bracket, spaces will be ignored after them. This is likely to lead to confusion as most users would not expect, say, `\gls{<label>}` to ignore following spaces, so `\new@ifnextchar` from the `amsgen` package is required.

The following keys can be used in the first optional argument. The `counter` key checks that the value is the name of a valid counter.

```
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
\def\@gls@counter{#1}%
```

```
}%
}
```

The value of the `format` key should be the name of a command (without the initial backslash) that has a single mandatory argument which can be used to format the associated entry number.

```
\define@key{glslink}{format}{%
\def\@glsnumberformat{#1}}
```

```
s@glslink@hyper@update@hook
```

```
\newcommand*{\@gls@glslink@hyper@update@hook}{}
```

The `hyper` key is a boolean key, it can either have the value `true` or `false`, and indicates whether or not to make a hyperlink to the relevant glossary entry. If `hyper` is `false`, an entry will still be made in the glossary, but the given text won't be a hyperlink.

```
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
```

Initialise `hyper` key.

```
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
```

Commands like `\gls` and `\glsdisp` unset the first use flag afterwards. Provide a way to localise this.

```
\@gls@save@glslocal
```

```
\newcommand*{\@gls@save@glslocal}{%
\let\if@org@KV@glslink@local\ifKV@glslink@local
}
```

```
\@gls@restore@glslocal
```

```
\newcommand*{\@gls@restore@glslocal}{%
\ifKV@glslink@local
\let\@gls@do@glsunset@glslocalunset
\else
\let\@gls@do@glsunset@glsunset
\fi
\let\ifKV@glslink@local\if@org@KV@glslink@local
}
```

```
\@gls@do@glsunset
```

```
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
```

The `local` key is a boolean key. If `true` this indicates that commands such as `\gls` should only do a local reset rather than a global one.

```
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
```

The original `\glsifhyper` command isn't particularly useful as it makes more sense to check the actual hyperlink setting rather than testing whether the starred or unstarred version has been used. Therefore, as from version

4.08, `\glsifhyper` is deprecated in favour of `\glsifhyperon`. In case there is a particular need to know whether the starred or unstarred version was used, provide a new command that determines whether the *-version, +-version or unmodified version was used.

```
\glslinkvar{<unmodified case>}{<star case>}{<plus case>}
```

`\glslinkvar` Initialise to unmodified case.

```
\newcommand*\glslinkvar}[3]{#1}
```

`\glsifhyper` `\glsifhyper` deprecated in v4.08 (2014-07-30) and removed in v4.50.

`\@gls@hyp@opt` Used by the commands such as `\glslink` to determine whether to modify the hyper option.

```
\newcommand*\@gls@hyp@opt}[1]{%
\let\glslinkvar\@firstofthree
\let\@gls@hyp@opt@cs#1\relax
\@ifstar{\s@gls@hyp@opt}%
{\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
```

`\s@gls@hyp@opt` Starred version

```
\newcommand*\s@gls@hyp@opt}[1] []{%
\let\glslinkvar\@secondofthree
\@gls@hyp@opt@cs[hyper=false,#1]}
```

`\p@gls@hyp@opt` Plus version

```
\newcommand*\p@gls@hyp@opt}[1] []{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
```

Syntax:

```
\glslink[<options>]{<label>}{<text>}
```

Display `<text>` in the document, and add the entry information for `<label>` into the relevant glossary. The optional argument should be a key value list using the `\glslink` keys defined above.

There is also a starred version:

```
\glslink* [<options>]{<label>}{<text>}
```

which is equivalent to `\glslink[hyper=false,<options>]{<label>}{<text>}`

First determine which version is being used:

```

\glslink
\newrobustcmd*{\glslink}{%
\@gls@hyp@opt\@gls@link
}

```

`\@gls@link` The main part of the business is in `\@gls@link` which shouldn't check if the term is defined as it's called by `\gls` etc which also perform that check.

```

\newcommand*{\@gls@link}[3] []{%
\glsdoifexistsordo{#2}%
{%
\let\do@gls@link@checkfirsthyper\relax
\@gls@link[#1]{#2}{#3}%
}%
}

```

Display the specified text. (The entry doesn't exist so there's nothing to link it to.)

```

\glstextformat{#3}%
}%

\glspostlinkhook
}

```

`\Glslink` Provide a sentence-case version in order to set up a map.

```

\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@link
}

\newcommand*{\@Gls@link}[3] []{%
\@gls@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}

```

`\glspostlinkhook`

```

\newcommand*{\glspostlinkhook}{}

```

`\@gls@link@checkfirsthyper` Check for first use and switch off hyper key if hyperlink not wanted. (Should be off if first use and `hyper=false` is on or if first use and both the entry is in an acronym list and the `acrfootnote` setting is on.) This assumes the glossary type is stored in `\glstype` and the label is stored in `\glslabel`.

```

\newcommand*{\@gls@link@checkfirsthyper}{%
\ifglsused{\glslabel}%
{%
}%
}%
{%
\gls@checkisacronymlist\glstype
\ifglshyperfirst
\if@glsisacronymlist
\ifglsacrfootnote
\KV@glslink@hyperfalse
}
}

```

```

        \fi
        \fi
    \else
        \KV@glslink@hyperfalse
    \fi
}%
Allow user to hook into this
    \glslinkcheckfirsthyperhook
}

\glslinkcheckfirsthyperhook Allow user to hook into the \@gls@link@checkfirsthyper macro
    \newcommand*\glslinkcheckfirsthyperhook{}

\glslinkpostsetkeys
    \newcommand*\glslinkpostsetkeys{}

\glsifhyperon Check the value of the hyper key:
    \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}

\do@glstdisablehyperinlist Disable hyperlink if in the “nohyper” list.
    \newcommand*\do@glstdisablehyperinlist{%
        \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
        {\KV@glslink@hyperfalse}{}%
    }

\gls@setdefault@glslink@opts Hook to set default options for \@glslink.
    \newcommand*\gls@setdefault@glslink@opts{}

\@gls@link
    \def\@gls@link[#1]#2#3{%
        Inserting \leavevmode suggested by Donald Arseneau (avoids problem with
        tabularx).
        \leavevmode
        \protected@edef\glslabel{\glsdetoklabel{#2}}%
        Save options in \@gls@link@opts and label in \@gls@link@label
        \def\@gls@link@opts{#1}%
        \let\@gls@link@label\glslabel
        \def\@glsnumberformat{glsnumberformat}%
        \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
        If this is in one of the “nohypertypes” glossaries, suppress the hyperlink by
        default
        \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
        Save original setting
        \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper

```

Save original setting of local key.

```
\@gls@save@glslocal
```

Set defaults:

```
\@gls@setdefault@glslink@opts
```

Switch off hyper setting if the glossary type has been identified in nohyperlist.

```
\do@gls@disablehyperinlist
```

Macros must set this before calling `\@gls@link`. The commands that check the first use flag should set this to `\@gls@link@checkfirsthyper` otherwise it should be set to `\relax`.

```
\do@gls@link@checkfirsthyper  
\setkeys{glslink}{#1}%
```

Add a hook for the user to customise things after the keys have been set.

```
\glslinkpostsetkeys
```

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

```
\@gls@setsort{glslabel}%
```

(De-tok'ing done by `\@do@wrglossary`)

```
\glsencapwrcontent{\@do@wrglossary{#2}}%  
\ifKV@glslink@hyper  
  \@glslink{glolinkprefixglslabel}{glsformat{#3}}%  
\else  
  \glsdonohyperlink{glolinkprefixglslabel}{glsformat{#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. Its purpose is to resolve [issue #189](#).

```
\newcommand*{\glsencapwrcontent}[1]{#1}
```

`\glolinkprefix`

```
\newcommand*{\glolinkprefix}{glo:}
```

`\glsentrycounter` Set default value of entry counter

```
\def\glsentrycounter{glscounter}%
```

`\@gls@saveentrycounter` Need to check if using equation counter in align environment:

```
\newcommand*{\@gls@saveentrycounter}{%  
  \def\@gls@Hcounter{}}%
```

Are we using equation counter?

```
\ifthenelse{\equal{\@gls@counter}{equation}}%  
{
```

If we're in align environment, `\xatlevel@` will be defined. (Can't test for `\@currentenv` as may be inside an inner environment.)

```
\ifcsundef{xatlevel@}%  
{%  
  \edef\theglentrycounter{\expandafter\noexpand  
    \csname the\@gls@counter\endcsname}%  
}%  
{%  
  \ifx\xatlevel@\@empty  
    \edef\theglentrycounter{\expandafter\noexpand  
      \csname the\@gls@counter\endcsname}%  
  \else  
    \savecounters@  
    \advance\c@equation by 1\relax  
    \edef\theglentrycounter{\csname the\@gls@counter\endcsname}%  
  \fi  
}%
```

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@mkidxrangear#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[#4]{#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
```

Check to see if the given string starts with a (or). If it does set `\@glo@prefix` to the starting character, and `\@glo@suffix` to the rest (or `glsnumberformat` if there is nothing else), otherwise set `\@glo@prefix` to nothing and `\@glo@suffix` to all of it.

```
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1\relax
  \def\@glo@prefix{(%}
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \if#1)\relax
    \def\@glo@prefix{)%}
    \if\relax#2\relax
      \def\@glo@suffix{glsnumberformat}%
    \else
      \def\@glo@suffix{#2}%
    \fi
  \else
    \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
  \fi
\fi}
```

`\@gls@escbsdq` Escape backslashes and double quote marks. The argument must be a control sequence.

```
\newcommand*\@gls@escbsdq[1]{%
  \def\@gls@checkedmkidx{%}
  \let\gls@xdystring=#1\relax
  \@onelevel@sanitize\gls@xdystring
  \edef\do@gls@xdycheckbackslash{%
```

```

\do@glsexdycheckbackslash\glsexdystring\noexpand@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@glsexdycheckbackslash
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{\glsexdystring}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexdycheckquote\glsexdystring\@nil""\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{\glsexdystring}%

```

Unsanitize \glsexnumberpage, \glsexalphpage, \glsexAlphpage and \glsexromanpage (thanks to David Carlisle for the suggestion.) Skip if esclocations=false.

```

\ifglsexesclocations
\@for\@glsex@tmp:=\glsexprotected@pagefmts\do
{%
\edef\@glsex@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@glsex@tmp}%
\@onelevel@sanitize\@glsex@sanitized@tmp
\edef\glsex@dosubst{%
\noexpand\DTLsubstituteall\noexpand\glsexdystring
{\@glsex@sanitized@tmp}{\expandonce\@glsex@tmp}%
}%
\glsex@dosubst
}%
\fi

```

Assign to required control sequence

```

\let#1=\glsexdystring
}

```

Catch special characters (argument must be a control sequence):

\@glsexcheckmkidxchars

```

\newcommand{\@glsexcheckmkidxchars}[1]{%
\ifglsexindy
\@glsexescbsdq{#1}%
\else
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckquote#1\@nil""\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckescquote#1\@nil\\"\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckescactual#1\@nil\?\?\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckactual#1\@nil??\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckbar#1\@nil||\null
\expandafter\@glsexupdatechecked\@glsexcheckedmkidx{#1}%
\def\@glsexcheckedmkidx{}%
\expandafter\@glsexcheckescbar#1\@nil\\|\null

```

```

\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%

\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesclevel#1\@nil!!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\edef\@do@gls@checkesc{%
  \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
  \@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}

```

Update the control sequence and strip trailing \@nil:

\@gls@updatechecked

```
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
```

\@gls@tmpb Define temporary token

```
\newtoks\@gls@tmpb
```

\@gls@checkquote Replace " with "" since " is a makeindex special character.

```

\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
  \def\@gls@checkquote{\relax}%
\else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
  \def\@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
  \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\@gls@quotechar}%
  \ifx\null#3\null
    \def\@gls@checkquote{\@gls@checkquote#2""\null}%
  \else
    \def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
  \fi
\fi
\@gls@checkquote
}

```

`\@gls@checkescquote` Do the same for `\`:

```
\def\@gls@checkescquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkescquote{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@quotechar
        \@gls@quotechar\string"\@gls@quotechar}%
      \def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@quotechar}%
    \ifx\null#3\null
      \def\@@gls@checkescquote{\@gls@checkescquote#2"\null}%
    \else
      \def\@@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
    \fi
  \fi
  \@@gls@checkescquote
}
```

`\@gls@checkescactual` Similarly for `\?` (which is replaces `@` as `makeindex`'s special character):

```
\def\@gls@checkescactual#1"?#2"?#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkescactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@actualchar
        \@gls@quotechar\string"\@gls@actualchar}%
      \def\@@gls@checkescactual{\@gls@checkescactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@actualchar}%
    \ifx\null#3\null
      \def\@@gls@checkescactual{\@gls@checkescactual#2"??\null}%
    \else
      \def\@@gls@checkescactual{\@gls@checkescactual#2"?#3\null}%
    \fi
  \fi
  \@@gls@checkescactual
}
```

```
}
```

`\@gls@checkescbar` Similarly for `\|`:

```
\def\@gls@checkescbar#1|#2|#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkescbar{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@encapchar
        \@gls@quotechar\string"\@gls@encapchar}%
      \def\@@gls@checkescbar{\@gls@checkescbar#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@encapchar}%
    \ifx\null#3\null
      \def\@@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
    \else
      \def\@@gls@checkescbar{\@gls@checkescbar#2|#3\null}%
    \fi
  \fi
  \@@gls@checkescbar
}
```

`\@gls@checkesclevel` Similarly for `\!`:

```
\def\@gls@checkesclevel#1#!#2#!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkesclevel{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string"\@gls@levelchar
        \@gls@quotechar\string"\@gls@levelchar}%
      \def\@@gls@checkesclevel{\@gls@checkesclevel#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\string"\@gls@levelchar}%
    \ifx\null#3\null
      \def\@@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
    \else
      \def\@@gls@checkesclevel{\@gls@checkesclevel#2#!#3\null}%
    \fi
}
```

```

\fi
\@gls@checkesclevel
}

```

\@gls@checkbar and for |:

```

\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}

```

\@gls@checklevel and for !:

```

\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi

```

```

\fi
\@@gls@checklevel
}

```

\@gls@checkactual and for ?:

```

\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@@gls@checkactual
}

```

\@gls@checkesc and for \:

```

\bgroup
\catcode'\|0\relax
\catcode'\|12\relax
\gdef|\@gls@checkesc#1\#2\#3\null{%
|\@gls@tmpb=\expandafter{|\@gls@checkedmkidx}%
\toks@={#1}%
\ifx|null#2|null
\ifx|null#3|null
\edef|\@gls@checkedmkidx{|\the|\@gls@tmpb|\the|\toks@}%
\def|\@@gls@checkesc{|\relax}%
\else
\edef|\@gls@checkedmkidx{|\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\@backslashchar|\@gls@quotechar|\@backslashchar}%
\def|\@@gls@checkesc{|\@gls@checkesc#3|null}%
\fi
\else
\edef|\@gls@checkedmkidx{|\the|\@gls@tmpb|\the|\toks@
|\@gls@quotechar|\@backslashchar}%
\ifx|null#3|null
\def|\@@gls@checkesc{|\@gls@checkesc#2\\|null}%

```

```

        |else
        |def|@gls@checkesc{|@gls@checkesc#2|#3|null}%
        |fi
        |fi
    |@gls@checkesc
}
|egroup

```

`\gls@xdycheckquote` As before but for use with xindy

```

\def@gls@xdycheckquote#1"#2"#3|null{%
@gls@tmpb=\expandafter{\gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef@gls@checkedmkidx{\the@gls@tmpb\the\toks@}%
\def@gls@xdycheckquote{\relax}%
\else
\edef@gls@checkedmkidx{\the@gls@tmpb\the\toks@
\string"\string"}%
\def@gls@xdycheckquote{\gls@xdycheckquote#3|null}%
\fi
\else
\edef@gls@checkedmkidx{\the@gls@tmpb\the\toks@
\string"}%
\ifx\null#3\null
\def@gls@xdycheckquote{\gls@xdycheckquote#2""\null}%
\else
\def@gls@xdycheckquote{\gls@xdycheckquote#2"#3\null}%
\fi
\fi
@gls@xdycheckquote
}

```

`\gls@xdycheckbackslash` Need to escape all backslashes for xindy. Define command that will define

```

@gls@xdycheckbackslash
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand@gls@tmpb=\noexpand\expandafter
{\noexpand@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand@gls@checkedmkidx{%
\noexpand\the\noexpand@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand@gls@checkedmkidx{%
\noexpand\the\noexpand@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%

```

```

\newlength\gls@tmpb
\newcommand\gls@xdi@checkbackslash[3]{%
  \noexpand\def\noexpand\@gls@xdi@checkbackslash{%
    \noexpand\@gls@xdi@checkbackslash##3\noexpand\null}%
  \noexpand\fi
\noexpand\else
\newcommand\gls@xdi@checkedmkidx[3]{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\newcommand\gls@xdi@checkbackslash[2]{%
  \noexpand\@gls@xdi@checkbackslash##2\@backslashchar
  \@backslashchar\noexpand\null}%
\noexpand\else
\newcommand\gls@xdi@checkbackslash[2]{%
  \noexpand\@gls@xdi@checkbackslash##2\@backslashchar
  ##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdi@checkbackslash
}%
}
Now go ahead and define \gls@xdi@checkbackslash
\def\gls@xdi@checkbackslash

```

`\glsdohypertarget`

```

\newlength\gls@tmpen
\newcommand\glsdohypertarget[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsmeasureheight{\gls@tmpen}{#2}%
    \glsdohypertargethook{#1}{#2}%
    \raisebox{\gls@tmpen}{\hypertarget{#1}{}}%
    #2%
  }%
}

```

`\glsdohypertargethook` Hook for `\glsdohypertarget`

```

\newcommand\glsdohypertargethook[2]{}

```

`\glslabelhypertarget` Provided for use in the definition of `\glsdohypertargethook` if required.

```

\NewDocumentCommand\glslabelhypertarget{mm}{%
  \protected@write\@auxout
  {\glslabelhypertargetdefs}%
  {%
    \string\newlabel
    {\glslabelhypertargetprefix#1}%
    {%
      {\glslabelhypertargetvalue}{\thepage}{#2}{#1}{}%
    }%
  }%
}

```

```

}
\ExplSyntaxOn

```

`\glslabelhypertargetvalue`

```

\newcommand\glslabelhypertargetvalue{
\bool_lazy_and:nnT
  { \tl_if_exist_p:N \glscurrententrylabel }
  { \bool_not_p:n { \tl_if_empty_p:N \glscurrententrylabel } }
{
  \glsentryname \glscurrententrylabel
}
}
\ExplSyntaxOff

```

`\glslabelhypertargetprefix`

```

\newcommand*\glslabelhypertargetprefix{}

```

`\glslabelhypertargetdefs`

```

\newcommand*\glslabelhypertargetdefs{%
\let\glossentryname\glsentryname
\let\glossentrydesc\glsentrydesc
\let\glossentrysymbol\glsentrysymbol
\let\Glossentryname\glsentryname
\let\Glossentrydesc\glsentrydesc
\let\Glossentrysymbol\glsentrysymbol
\let\glossentrynameother\@gls@entry@field
}

```

`\glsdohyperlink`

```

\newcommand*\glsdohyperlink}[2]{%
\glsdoshowtarget{#1}%
{%
  \glsdohyperlinkhook{#1}{#2}%
  \hyperlink{#1}{#2}%
}%
}

```

`\glsdohyperlinkhook` Hook for `\glsdohyperlink`

```

\newcommand*\glsdohyperlinkhook}[2]{}

```

`\glsdonohyperlink`

```

\newcommand*\glsdonohyperlink}[2]{#2}

```

`\@glslink` If `\hyperlink` is not defined `\@glslink` ignores its first argument and just does the second argument, otherwise it is equivalent to `\hyperlink`.

```

\ifcsundef{hyperlink}%
{%

```

```

\let\@glslink\glsdonohyperlink
}%
{%
\let\@glslink\glsdohyperlink
}

```

`\@glstarget` If `\hypertarget` is not defined, `\@glstarget` ignores its first argument and just does the second argument, otherwise it is equivalent to `\hypertarget`.

```

\ifcsundef{hypertarget}%
{%
\let\@glstarget\@secondoftwo
}%
{%
\let\@glstarget\glsdohypertarget
}

```

Glossary hyperlinks can be disabled using `\glsdisablehyper` (effect can be localised):

`\glsdisablehyper`

```

\newcommand{\glsdisablehyper}{%
\KV@glslink@hyperfalse
\let\@glslink\glsdonohyperlink
\let\@glstarget\@secondoftwo
}

```

Glossary hyperlinks can be enabled using `\glsenablehyper` (effect can be localised):

`\glsenablehyper`

```

\newcommand{\glsenablehyper}{%
\KV@glslink@hypertrue
\let\@glslink\glsdohyperlink
\let\@glstarget\glsdohypertarget
}

```

Provide some convenience commands if not already defined:

```

\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}

```

Syntax:

```

\gls[options]{label}[insert text]

```

Link to glossary entry using singular form. The link text is taken from the value of the text or first keys used when the entry was defined.

The first optional argument is a key-value list, the same as `\glslink`, the mandatory argument is the entry label. After the mandatory argument, there is another optional argument to insert extra text in the link text (the location

of the inserted text is governed by `\glsdisplay` and `\glsdisplayfirst`). As with `\glslink` there is a starred version which is the same as the unstarred version but with the `hyper` key set to `false`. (Additional options can also be specified in the first optional argument.)

First determine which version is being used:

```
\gls
  \newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\@gls
  \newcommand*{\@gls}[2][\@gls@hyp@opt\@gls]{%
    \new@ifnextchar[{\@gls@#1}{#2}]{\@gls@#1}{#2}[1]}%
  }
```

`\@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\@firstofthree
    \let\glsifcustomtext\@empty
    \def\glsinsert{#3}%
  }
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\gls@type`.

```
\def\@glo@text{\csname gls@\gls@type @entryfmt\endcsname}
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\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@gls@unset{#2}%
}%

\gls@postlinkhook
}
```

`\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}
```



```

\let\glsifplural\@secondoftwo
\let\glsifcaps\@thirdofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%

```

Determine what the link text should be (this is stored in `\@glo@text`). Note that `\@gls@link` sets `\glstyp`.

```

\def\@glo@text{\csname gls@\glstyp @entryfmt\endcsname}%

```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```

\@gls@link[#1]{#2}{\@glo@text}%

```

Indicate that this entry has now been used

```

\@gls@do@glssunset{#2}%
}%

\glspostlinkhook
}

```

`\glspl` behaves in the same way as `\gls` except it uses the plural form.

`\glspl`

```

\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}

```

Defined the un-starred form. Need to determine if there is a final optional argument

```

\newcommand*{\@glspl}[2] [] {%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}

```

`\@glspl@` Read in the final optional argument:

```

\def\@glspl@#1#2[#3] {%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

\let\glsifplural\@firstoftwo
\let\glsifcaps\@firstofthree
\let\glsifcustomtext\@empty
\def\glsinsert{#3}%

```

Determine what the link text should be (this is stored in `\@glo@text`). Note that `\@gls@link` sets `\glstyp`.

```

\def\@glo@text{\csname gls@\glstyp @entryfmt\endcsname}%

```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```

\@gls@link[#1]{#2}{\@glo@text}%

```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

`\Glspl` behaves in the same way as `\glspl`, except that the first letter of the link text is converted to uppercase (as with `\Gls`, if the first letter has an accent, it will need to be grouped).

`\Glspl`

```
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glspl}{\Glspl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glspl}[2] [] {%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
```

`\@Glspl@` Read in the final optional argument:

```
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }
```

Determine what the link text should be (this is stored in `\@glo@text`). This needs to be expanded so that the `\@glo@text` can be passed to `\xmakefirstuc`. Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%

\glspostlinkhook
}
```

`\GLSp1` behaves like `\glsp1` except that all the link text is converted to uppercase.

`\GLSp1`

```
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSp1}[2] []{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2} []}%
}
```

`\@GLSp1` Read in the final optional argument:

```
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper

    \let\glsifplural\@firstoftwo
    \let\glsescapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%
```

```
\glspostlinkhook
}
```

`\glsdisp` `\glsdisp[options]{label}{text}` This is like `\gls` except that the link text is provided. This differs from `\glslink` in that it uses `\glsdisplay` or `\glsdisplayfirst` and unsets the first use flag.

First determine if we are using the starred form:

```
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
```

Defined the un-starred form.

`\@glsdisp`

```
\newcommand*{\@glsdisp}[3] []{%
  \glsdoifexists{#2}{%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

```
\let\glsifplural\@secondoftwo
\let\glsapscase\@firstofthree
\def\glscustomtext{#3}%
\def\glsinsert{}%
```

Determine what the link text should be (this is stored in `\@glo@text`) Note that `\@gls@link` sets `\glstype`.

```
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
```

Call `\@gls@link`. If footnote package option has been used and the glossary type is `\acronymtype`, suppress hyperlink for first use. Likewise if the `hyperfirst=false` package option is used.

```
\@gls@link[#1]{#2}{\@glo@text}%
```

Indicate that this entry has now been used

```
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
```

`\Glsdisp` Provide a sentence-case version to make it easier to map.

```
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
```

`\@Glsdisp`

```
\newcommand*{\@Glsdisp}[3][{}]{%
\@glsdisp[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glsdisp}{\@Glsdisp}
```

`\@gls@link@nocheckfirsthyper` Instead of just setting `\do@gls@link@checkfirsthyper` to `\relax` in `\@gls@field@link`, set it to `\@gls@link@nocheckfirsthyper` in case some other action needs to take place.

```
\newcommand*{\@gls@link@nocheckfirsthyper}{}
```

```
\@gls@field@link{\langle options \rangle}{\langle label \rangle}{\langle link text \rangle}
```

`\@gls@field@link`

```
\newcommand{\@gls@field@link}[3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\@gls@link[#1]{#2}{#3}%
}%
\glspostlinkhook
}
```

`\glstext` behaves like `\gls` except it always uses the value given by the text key and it doesn't mark the entry as used.

`\glstext`

```
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glstext}[2] []{%  
  \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glstext@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%  
}
```

`\GLStext` behaves like `\glstext` except the text is converted to uppercase.

`\GLStext`

```
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}  
\glsmfublocker{\GLStext}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLStext}[2] []{%  
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLStext@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrytext{#2}#3}}%  
}
```

`\Glstext` behaves like `\glstext` except that the first letter of the text is converted to uppercase.

`\Glstext`

```
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glstext}{\Glstext}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glstext}[2] []{%  
  \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glstext@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%  
}
```

`\glsfirst` behaves like `\gls` except it always uses the value given by the first key and it doesn't mark the entry as used.

`\glsfirst`

```
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsfirst}[2] [] {%  
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []]}
```

Read in the final optional argument:

```
\def\@glsfirst@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%  
}
```

`\Glsfirst` behaves like `\glsfirst` except it displays the first letter in uppercase.

`\Glsfirst`

```
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirst}{\Glsfirst}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsfirst}[2] [] {%  
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []]}
```

Read in the final optional argument:

```
\def\@Glsfirst@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%  
}
```

`\GLSfirst` behaves like `\Glsfirst` except it displays the text in uppercase.

`\GLSfirst`

```
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}  
\glsmfublocker{\GLSfirst}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSfirst}[2] [] {%  
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []]}
```

Read in the final optional argument:

```
\def\@GLSfirst@#1#2[#3] {%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryfirst{#2}#3}%  
}
```

`\glsplural` behaves like `\gls` except it always uses the value given by the plural key and it doesn't mark the entry as used.

`\glsplural`

```
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@glsplural}[2] []{%  
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%  
}
```

`\Glsplural` behaves like `\glsplural` except that the first letter is converted to uppercase.

`\Glsplural`

```
\newrobustcmd*\@Glsplural}{\@gls@hyp@opt\@Glsplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsplural}{\Glsplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@Glsplural}[2] []{%  
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%  
}
```

`\GLSplural` behaves like `\glsplural` except that the text is converted to uppercase.

`\GLSplural`

```
\newrobustcmd*\@GLSplural}{\@gls@hyp@opt\@GLSplural}  
\glsmfublocker{\GLSplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@GLSplural}[2] []{%  
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryplural{#2}#3}}%  
}
```

`\glsfirstplural` behaves like `\gls` except it always uses the value given by the `firstplural` key and it doesn't mark the entry as used.

`\glsfirstplural`

```
\newrobustcmd*\@glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsfirstplural}[2] []{%  
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glstentryfirstplural{#2}#3}%  
}
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

```
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSfirstplural}[2] []{%  
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\GLStentryfirstplural{#2}#3}%  
}
```

`\GLSfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\GLSfirstplural`

```
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}  
\glsmfublocker{\GLSfirstplural}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSfirstplural}[2] []{%  
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glstentryfirstplural{#2}#3}}%  
}
```

`\glsname` behaves like `\gls` except it always uses the value given by the name key and it doesn't mark the entry as used.

`\glsname`

```
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsname}[2] [] {%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
```

`\Glsname` behaves like `\glsname` except that the first letter is converted to uppercase.

`\Glsname`

```
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsname}{\Glsname}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSname}[2] [] {%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\GLSentryname{#2}#3}%
}
```

`\GLSname` behaves like `\glsname` except that the link text is converted to uppercase.

`\GLSname`

```
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\glsmfublocker{\GLSname}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSname}[2] [] {%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryname{#2}#3}}%
}
```

`\glsdesc` behaves like `\gls` except it always uses the value given by the description key and it doesn't mark the entry as used.

`\glsdesc`

```
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\glsdesc}[2] [] {%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
```

\Glsdesc behaves like \glsdesc except that the first letter is converted to uppercase.

\Glsdesc

```
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsdesc}{\Glsdesc}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsdesc}[2] [] {%
  \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
```

\GLSdesc behaves like \glsdesc except that the link text is converted to uppercase.

\GLSdesc

```
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\glsmfublocker{\GLSdesc}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%
}
```

\glsdescplural behaves like \gls except it always uses the value given by the descriptionplural key and it doesn't mark the entry as used.

\glsdescplural

```
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\glsdescplural}[2] [] {%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
```

`\Glsdescplural` behaves like `\glsdescplural` except that the first letter is converted to uppercase.

`\Glsdescplural`

```
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
```

`\GLSdescplural` behaves like `\glsdescplural` except that the link text is converted to uppercase.

`\GLSdescplural`

```
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydescplural{#2}#3}}%
}
```

`\glsymbol` behaves like `\gls` except it always uses the value given by the symbol key and it doesn't mark the entry as used.

`\glsymbol`

```
\newrobustcmd*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\glssymbol}[2] []{%
  \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
}
```

\Glssymbol behaves like \glssymbol except that the first letter is converted to uppercase.

\Glssymbol

```
\newrobustcmd*{\Glssymbol}{\@gls@hyp@opt\@Glssymbol}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbol}{\Glssymbol}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@Glssymbol}[2] []{%
  \new@ifnextchar[{\@Glssymbol@{#1}{#2}}{\@Glssymbol@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\GLstentrysymbol{#2}#3}%
}
```

\GLSsymbol behaves like \glssymbol except that the link text is converted to uppercase.

\GLSsymbol

```
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\glsmfublocker{\GLSsymbol}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSsymbol}[2] []{%
  \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glstentrysymbol{#2}#3}}%
}
```

\glsymbolplural behaves like \gls except it always uses the value given by the symbolplural key and it doesn't mark the entry as used.

\glsymbolplural

```
\newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glssymbolplural}[2] [] {%
  \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
}
```

`\Glssymbolplural` behaves like `\glssymbolplural` except that the first letter is converted to uppercase.

`\Glssymbolplural`

```
\newrobustcmd*{\Glssymbolplural}{\@gls@hyp@opt\@Glssymbolplural}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glssymbolplural}{\Glssymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\GLStentrysymbolplural{#2}#3}%
}
```

`\GLSsymbolplural` behaves like `\glssymbolplural` except that the link text is converted to uppercase.

`\GLSsymbolplural`

```
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@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{\glstentrysymbolplural{#2}#3}}%
}
```

`\glsuseri` behaves like `\gls` except it always uses the value given by the `user1` key and it doesn't mark the entry as used.

`\glsuseri`

```
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@glsuseri}[2] []{%  
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuseri@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%  
}
```

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

```
\newrobustcmd*\@Glsuseri{\@gls@hyp@opt\@Glsuseri}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseri}{\@Glsuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@GLSuseri}[2] []{%  
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuseri@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\GLSentryuseri{#2}#3}%  
}
```

`\GLSuseri` behaves like `\glsuseri` except that the link text is converted to uppercase.

`\GLSuseri`

```
\newrobustcmd*\@GLSuseri{\@gls@hyp@opt\@GLSuseri}  
\glsmfublocker{\@GLSuseri}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@GLSuseri}[2] []{%  
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuseri@#1#2[#3]{%  
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}%  
}
```

`\glsuserii` behaves like `\gls` except it always uses the value given by the `user2` key and it doesn't mark the entry as used.

`\glsuserii`

```
\newrobustcmd*\@glsuserii{\@gls@hyp@opt\@glsuserii}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@glsuserii}[2] []{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
```

\Glsuserii behaves like \glsuserii except that the first letter is converted to uppercase.

\Glsuserii

```
\newrobustcmd*\@Glsuserii{\@gls@hyp@opt\@Glsuserii}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserii}{\Glsuserii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@Glsuserii}[2] []{%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
```

\GLSuserii behaves like \glsuserii except that the link text is converted to uppercase.

\GLSuserii

```
\newrobustcmd*\@GLSuserii{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@GLSuserii}[2] []{%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
```

\glsuseriii behaves like \gls except it always uses the value given by the user3 key and it doesn't mark the entry as used.

\glsuseriii

```
\newrobustcmd*\@glsuseriii{\@gls@hyp@opt\@glsuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@glsuseriii[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
```

`\Glsuseriii` behaves like `\glsuseriii` except that the first letter is converted to uppercase.

`\Glsuseriii`

```
\newrobustcmd*\@Glsuseriii{\@gls@hyp@opt\@Glsuseriii}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@GLsuseriii[2] [] {%
  \new@ifnextchar[{\@GLsuseriii@{#1}{#2}}{\@GLsuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@GLsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\GLsentryuseriii{#2}#3}%
}
```

`\GLSuseriii` behaves like `\glsuseriii` except that the link text is converted to uppercase.

`\GLSuseriii`

```
\newrobustcmd*\@GLSuseriii{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*\@GLSuseriii[2] [] {%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
```

`\glsuseriv` behaves like `\gls` except it always uses the value given by the `user4` key and it doesn't mark the entry as used.

`\glsuseriv`

```
\newrobustcmd*\@glsuseriv{\@gls@hyp@opt\@glsuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuseriv}[2] []{%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
```

`\Glsuseriv` behaves like `\glsuseriv` except that the first letter is converted to uppercase.

`\Glsuseriv`

```
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseriv}[2] []{%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\GLSentryuseriv{#2}#3}%
}
```

`\GLSuseriv` behaves like `\glsuseriv` except that the link text is converted to uppercase.

`\GLSuseriv`

```
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuseriv}[2] []{%
  \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
```

`\glsuserv` behaves like `\gls` except it always uses the value given by the `user5` key and it doesn't mark the entry as used.

`\glsuserv`

```
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuserv}[2][\%
  \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
```

\Glsuserv behaves like \glsuserv except that the first letter is converted to uppercase.

\Glsuserv

```
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserv}{\Glsuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuserv}[2][\%
  \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserv{#2}#3}%
}
```

\GLSuserv behaves like \glsuserv except that the link text is converted to uppercase.

\GLSuserv

```
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSuservi}[2][\%
  \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]}
```

Read in the final optional argument:

```
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}%
}
```

\glsuservi behaves like \gls except it always uses the value given by the user6 key and it doesn't mark the entry as used.

\glsuservi

```
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@glsuservi}[2] []{%
  \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

```
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsuservi}{\Glsuservi}
```

Defined the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLsuservi}[2] []{%
  \new@ifnextchar[{\@GLsuservi@{#1}{#2}}{\@GLsuservi@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\GLsentryuservi{#2}#3}%
}
```

\GLSuservi behaves like \glsuservi except that the link text is converted to uppercase.

\GLSuservi

```
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfublocker{\GLSuservi}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@GLSsuservi}[2] []{%
  \new@ifnextchar[{\@GLSsuservi@{#1}{#2}}{\@GLSsuservi@{#1}{#2} []}}
```

Read in the final optional argument:

```
\def\@GLSsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
```

Now deal with acronym related keys. First the short form:

\acrshort

```
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} [] }%
}
```

Read in the final optional argument:

```
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
  }
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstyp`.

```
\@gls@link[#1]{#2}{\csname gls@\glstyp @entryfmt\endcsname}%
}%

\gls post link hook
}
```

`\Acrshort`

```
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrshort}{\Acrshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} [] }%
}
```

Read in the final optional argument:

```
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
  }
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstyp`.

```
\@gls@link[#1]{#2}{\csname gls@\glstyp @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\ACRshort`

```
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
```

Read in the final optional argument:

```
\def\@ACRshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsentryshort{#2}}#3}%
    }%
  }%
}
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstyp`.

```
\@gls@link[#1]{#2}{\csname gls@\glstyp @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

Short plural:

`\acrshortpl`

```
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
}
```

Read in the final optional argument:

```
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshortpl{#2}}#3%
    }%
  }
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}
```

`\Acrshortpl`

```
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\ns@Acrshortpl}[2][ ]{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} [ ]}%
}
```

Read in the final optional argument:

```
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapspace\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
  }
```

Call `\@gls@link` Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
```

```

    \glspostlinkhook
  }

\ACRshortpl
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}

Define the un-starred form. Need to determine if there is a final optional argu-
ment
\newcommand*{\ns@ACRshortpl}[2] [] {%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}]%
}

Read in the final optional argument:
\def\@ACRshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsupercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
  }%

Call \@gls@link Note that \@gls@link sets \gls type.
  \@gls@link{#1}{#2}{\csname gls@gls type @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

\acrlong
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}

Define the un-starred form. Need to determine if there is a final optional argu-
ment
\newcommand*{\ns@acrlong}[2] [] {%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}]%
}

Read in the final optional argument:
\def\@acrlong#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\gls caps case\@firstofthree
\let\glsinsert\@empty

```

Bug fix v4.02 removed `\acronymfont` from `\gls customtext` (`\acronymfont` only designed for short form).

```

\def\gls customtext{%
  \glsentrylong{#2}#3%
}%

```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```

\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\gls post link hook
}

```

`\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\gls caps case\@secondofthree
    \let\glsinsert\@empty

```

Bug fix v4.02 removed `\acronymfont` from `\gls customtext` (`\acronymfont` only designed for short form).

```

\def\gls customtext{%
  \Glsentrylong{#2}#3%
}%

```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```

\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

```

```

    \glspostlinkhook
  }
\ACRlong
  \newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
  \glsmfublocker{\ACRlong}

Define the un-starred form. Need to determine if there is a final optional argument
\newcommand*{\ns@ACRlong}[2] [] {%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2} []}]%
}

Read in the final optional argument:
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont
only designed for short form).
    \def\glscustomtext{%
      \glsuppercase{\glsentrylong{#2}#3}%
    }%

Call \@gls@link. Note that \@gls@link sets \glstype.
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

Short plural:
\acrlongpl
  \newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}

Define the un-starred form. Need to determine if there is a final optional argument
\newcommand*{\ns@acrlongpl}[2] [] {%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2} []}]%
}

Read in the final optional argument:
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%  
\let\glsifplural\@firstoftwo  
\let\glsifcaps\@firstofthree  
\let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%  
  \glsentrylongpl{#2}#3%  
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%  
}%  
  
\glspostlinkhook  
}
```

`\Acrlongpl`

```
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
```

Add sentence case mapping.

```
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
```

Define the un-starred form. Need to determine if there is a final optional argument

```
\newcommand*{\@ns@Acrlongpl}[2][{}]{%  
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%  
}
```

Read in the final optional argument:

```
\def\@Acrlongpl#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%  
\let\glsifplural\@firstoftwo  
\let\glsifcaps\@secondofthree  
\let\glsinsert\@empty
```

Bug fix v4.02 removed `\acronymfont` from `\glscustomtext` (`\acronymfont` only designed for short form).

```
\def\glscustomtext{%  
  \Glsentrylongpl{#2}#3%  
}%
```

Call `\@gls@link`. Note that `\@gls@link` sets `\glstype`.

```
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%  
}%
```

```

    \glspostlinkhook
  }

\ACRlongpl
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}

Define the un-starred form. Need to determine if there is a final optional argu-
ment
\newcommand*{\ns@ACRlongpl}[2][ ]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2} [ ]}%
}

Read in the final optional argument:
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty

Bug fix v4.02 removed \acronymfont from \glscustomtext (\acronymfont
only designed for short form).
    \def\glscustomtext{%
      \glsuppercase{\glsentrylongpl{#2}{#3}}%
    }%

Call \@gls@link. Note that \@gls@link sets \glsstyle.
    \@gls@link[#1]{#2}{\csname gls@\glsstyle @entryfmt\endcsname}%
  }%

  \glspostlinkhook
}

```

2.11.2 Displaying entry details without adding information to the glossary

These commands merely display entry information without adding entries in the associated file or having hyperlinks.

`\@gls@entry@field` Generic version.

```
\@gls@entry@field{\label}{\field}
```

Now using `\csuse` instead of `\csname` to expand to nothing rather than `\relax` if undefined.

```
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
```

`\glsletentryfield`

```
\glsletentryfield{<cs>}{<label>}{<field>}
```

```
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
```

`\@Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{<label>}{<field>}
```

Check for PDF bookmarks performed by `\@Glsentryfield`.

```
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\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`

```
\@Glsentryfield{<entry-label>}{<field-label>}
```

```
\newcommand{\@Glsentryfield}[2]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
```

`\glstexorpdfstring` Provide a command that will use `\texorpdfstring` if it has been defined. (Remove all the cumbersome tests to determine whether or not that command has been defined.)

```
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
```

Get the entry name (as specified by the `name` key when the entry was defined). The argument is the label associated with the entry. Note that unless you used `name=false` in the `sanitize` package option you may get unexpected results if the `name` key contains any commands.

`\glsentryname`

```
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
```

The sentence-case equivalents were made robust since they use the non-expandable `\makefirstuc`. These will now expand in the PDF bookmarks to sentence case using `\MFUsentencecase` (which is expandable). Obviously, if any of the fields contain non-expandable content (such as acronym font changing commands) then that will cause a problem (which can be dealt with by using `hyperref`'s `\pdfstringdefDisableCommands`).

`\Glsentryname`

```
\newcommand*{\Glsentryname}[1]{%
\glstexorpdfstring{\protect\@Gls@entryname{#1}}%
{\MFUsentencecase{\@gls@entry@field{#1}{name}}}%
}

\glsmfuexcl{\Glsentryname}
```

`\@Gls@entryname`

This is a workaround in the event that the user defies the warning in the manual about not using `\Glsname` or `\Glsentryname` with acronyms. The check for PDF bookmarks is in `\Glsentryname`. First the default behaviour:

```
\newcommand*{\@Gls@entryname}[1]{%
\@Gls@entry@field{#1}{name}%
}
```

`\@Gls@acentryname`

Now the behaviour when `\setacronymstyle` is used. The original definition of this command used `mfirstuc` internal commands that are now being deprecated.

```
\newrobustcmd*{\@Gls@acentryname}[1]{%
\ifglshaslong{#1}%
{%
```

Just need to ensure that `\acrnameformat` is expanded, which may cause problems as non-robust semantic commands may expand to content that interferes with sentence-casing.

```
\protected@edef\@glo@text{\@gls@entry@field{#1}{name}}%
\expandafter\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@gls@sentencecase\expandafter{@glo@text}%  
  \fi  
  \fi  
  \fi  
  \fi  
}%  
{%
```

Not an acronym

```
\@Gls@entry@field{#1}{name}%  
}%  
}
```

Get the entry description (as specified by the `description` key when the entry was defined). The argument is the label associated with the entry. Note that unless you used `description=false` in the `sanitize` package option you may get unexpected results if the `description` key contained any commands.

```
\glsentrydesc
  \newcommand*\glsentrydesc[1]{\@gls@entry@field{#1}{desc}}
```

```
\Glsentrydesc
  \newcommand*\Glsentrydesc[1]{\@Glsentryfield{#1}{desc}}
```

Version 4.54: sentence case mapping removed as `\glsentrydesc` is expandable. This ensures that `\glsentrydesc` can be converted to all-caps if required. However, `\Glsentrydesc` needs to be added as an exclusion. (Similarly for the other `\Glsentry<field>` commands)

```
\glsmfuexcl{\Glsentrydesc}
```

Plural form:

```
\glsentrydescplural
  \newcommand*\glsentrydescplural[1]{%
    \@gls@entry@field{#1}{descplural}%
  }
```

```
\Glsentrydescplural
  \newcommand*\Glsentrydescplural[1]{%
    \@Glsentryfield{#1}{descplural}%
  }
```

```
\glsmfuexcl{\Glsentrydescplural}
```

Get the entry text, as specified by the `text` key when the entry was defined. The argument is the label associated with the entry:

```
\glsentrytext
  \newcommand*\glsentrytext[1]{\@gls@entry@field{#1}{text}}
```

```
\Glsentrytext
  \newcommand*\Glsentrytext[1]{%
    \@Glsentryfield{#1}{text}%
  }
```

```
\glsmfuexcl{\Glsentrytext}
```

Get the plural form:

```
\glsentryplural
  \newcommand*\glsentryplural[1]{%
    \@gls@entry@field{#1}{plural}%
  }
```

`\Glsentryplural`

```
\newcommand*\Glsentryplural}[1]{%  
  \@Glsentryfield{#1}{plural}%  
}
```

```
\glsmfuexcl{\Glsentryplural}
```

Get the symbol associated with this entry. The argument is the label associated with the entry.

`\glsentrysymbol`

```
\newcommand*\glsentrysymbol}[1]{%  
  \@gls@entry@field{#1}{symbol}%  
}
```

`\Glsentrysymbol`

```
\newcommand*\Glsentrysymbol}[1]{%  
  \@Glsentryfield{#1}{symbol}%  
}
```

```
\glsmfuexcl{\Glsentrysymbol}
```

Plural form:

`\glsentrysymbolplural`

```
\newcommand*\glsentrysymbolplural}[1]{%  
  \@gls@entry@field{#1}{symbolplural}%  
}
```

`\Glsentrysymbolplural`

```
\newcommand*\Glsentrysymbolplural}[1]{%  
  \@Glsentryfield{#1}{symbolplural}%  
}
```

```
\glsmfuexcl{\Glsentrysymbolplural}
```

Get the entry text to be used when the entry is first used in the document (as specified by the first key when the entry was defined).

`\glsentryfirst`

```
\newcommand*\glsentryfirst}[1]{%  
  \@gls@entry@field{#1}{first}%  
}
```

`\Glsentryfirst`

```
\newcommand*\Glsentryfirst}[1]{%  
  \@Glsentryfield{#1}{first}%  
}
```

```
\glsmfuexcl{\Glsentryfirst}
```

Get the plural form (as specified by the `firstplural` key when the entry was defined).

```
\glsentryfirstplural
  \newcommand*\glsentryfirstplural}[1]{%
    \@gls@entry@field{#1}{firstpl}%
  }
```

```
\Glsentryfirstplural
  \newcommand*\Glsentryfirstplural}[1]{%
    \@Glsentryfield{#1}{firstpl}%
  }

  \glsmfuexcl{\Glsentryfirstplural}
```

`\glscapitalisewords` Make it easier to switch from `\capitalisewords` to `\capitalisefmtwords`.

```
\newcommand{\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
```

`\glsentrytitlecase` Now uses sentence case in PDF bookmark.

```
\newcommand*\glsentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
```

```
\@glsentrytitlecase
  \newrobustcmd*\@glsentrytitlecase}[2]{%
    \glsdoifexists{#1}%
    {%
      \glsfieldfetch{#1}{#2}{\@gls@value}%
      \expandafter\glscapitalisewords\expandafter{\@gls@value}%
    }%
  }
```

Display the glossary type with which this entry is associated (as specified by the `type` key used when the entry was defined). This expands to a label so isn't intended to produce text in the document.

```
\glsentrytype
  \newcommand*\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
```

Display the sort text used for this entry. Note that the `sort` key is `sanitize`, so unexpected results may occur if the `sort` key contained commands. This is likewise not intended to produce text in the document, other than for debugging purposes.

```

\glsentrysort
    \newcommand*\glsentrysort[1]{%
        \gls@entry@field{#1}{sort}%
    }

\glsentryparent Expands to the label of the entry's parent. This expands to a label so isn't
intended to produce text in the document.
    \newcommand*\glsentryparent[1]{%
        \gls@entry@field{#1}{parent}%
    }

\glsentryuseri Get the first user key (as specified by the user1 when the entry was defined).
The argument is the label associated with the entry.
    \newcommand*\glsentryuseri[1]{%
        \gls@entry@field{#1}{useri}%
    }

\Glsentryuseri
    \newcommand*\Glsentryuseri[1]{%
        \Glsentryfield{#1}{useri}%
    }

    \glsmfuexcl{\Glsentryuseri}

\glsentryuserii Get the second user key (as specified by the user2 when the entry was defined).
The argument is the label associated with the entry.
    \newcommand*\glsentryuserii[1]{%
        \gls@entry@field{#1}{userii}%
    }

\Glsentryuserii
    \newcommand*\Glsentryuserii[1]{%
        \Glsentryfield{#1}{userii}%
    }

    \glsmfuexcl{\Glsentryuserii}

\glsentryuseriii Get the third user key (as specified by the user3 when the entry was defined).
The argument is the label associated with the entry.
    \newcommand*\glsentryuseriii[1]{%
        \gls@entry@field{#1}{useriii}%
    }

\Glsentryuseriii
    \newcommand*\Glsentryuseriii[1]{%
        \Glsentryfield{#1}{useriii}%
    }

```

Add sentence case mapping.

```
\glsmfuexcl{\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}%
}
```

```
\glsmfuexcl{\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}%
}
```

```
\glsmfuexcl{\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}%
}
```

```
\glsmfuexcl{\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}%
  }

  \glsmfuexcl{\Glsentryshort}

\glsentryshortpl Get the short plural key (as specified by the shortplural the entry was defined).
The argument is the label associated with the entry.
  \newcommand*\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}

\Glsentryshortpl
  \newcommand*\Glsentryshortpl}[1]{%
    \@Glsentryfield{#1}{shortpl}%
  }

  \glsmfuexcl{\Glsentryshortpl}

\glsentrylong Get the long key (as specified by the long the entry was defined). The argument
is the label associated with the entry.
  \newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}

\Glsentrylong
  \newcommand*\Glsentrylong}[1]{%
    \@Glsentryfield{#1}{long}%
  }

  \glsmfuexcl{\Glsentrylong}

\glsentrylongpl Get the long plural key (as specified by the longplural the entry was defined).
The argument is the label associated with the entry.
  \newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}

\Glsentrylongpl
  \newcommand*\Glsentrylongpl}[1]{%
    \@Glsentryfield{#1}{longpl}%
  }

  \glsmfuexcl{\Glsentrylongpl}

  Short cut macros to access full form:

\glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
  \newcommand*\glsentryfull}[1]{%
    \glsentrylong{#1} (\glsentryshort{#1})%
  }

```

`\Glsentryfull` This definition shouldn't actually be used. This command will be redefined by the acronym style.

```
\newcommand*\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glsentryshort{#1})%
}

\glsmfexcl{\Glsentryfull}
```

`\glsentryfullpl` This definition shouldn't actually be used. This command will be redefined by the acronym style.

```
\newcommand*\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
```

`\Glsentryfullpl` This definition shouldn't actually be used. This command will be redefined by the acronym style.

```
\newcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
%\changes{4.54}{2024-04-03}{change map to exclusion}

\glsmfexcl{\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@glsglnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
  \csname glo@\glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
  }%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}

```

`\glsnumlistsep`

```
\newcommand*{\glsnumlistsep}{, }
```

`\glsnumlistlastsep`

```
\newcommand*{\glsnumlistlastsep}{ \& }
```

`\glshyperlink` Provide a hyperlink to a glossary entry without adding information to the glossary file. The entry needs to be added using a command like `\glslink` or `\glsadd` to ensure that the target is defined. The first (optional) argument specifies the link text. The entry name is used by default. The second argument is the entry label.

```

\newcommand*{\glshyperlink}[2][\glsentrytext{\glo@label}]{%
  \def\glo@label{#2}%
  \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}

```

2.12 Adding an entry to the glossary without generating text

The following keys are provided for `\glsadd` and `\glsaddall`:

```
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
```

This key is only used by `\glsaddall`:

```
\define@key{glossadd}{types}{\def\@glo@type{#1}}
```

```
\glsadd[<options>]{<label>}
```

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that `<options>` only has two keys: `counter` and `format` (the `types` key will be ignored).

`\glsadd`

```
\newrobustcmd*{\glsadd}[2] []{%
```

Need to move to horizontal mode if not already in it, but only if not in preamble.

```
\@gls@adjustmode
\glsdoifexists{#2}%
{%
\def\@glsnumberformat{glsnumberformat}%

\protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
\setkeys{glossadd}{#1}%
```

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

```
\@gls@setsort{#2}%
```

This should use `\@do@wrglossary` rather than `\do@wrglossary` since the whole point of `\glsadd` is to add a line to the glossary.

```
\@do@wrglossary{#2}%
}%
}
```

Identify `\glsadd` as an exclusion, but this will only work if the optional argument is omitted. If the optional argument is required, the `\glsadd` command with arguments will need to be encapsulated with `\MFUskipunc`. See the `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][\@glo@types]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
```

```
\glsaddallunused[glossary types]
```

```
\glsaddallunused
```

Add all used terms defined for the listed glossaries (without displaying any text). If optional argument is omitted, apply to all glossary types. This should typically go at the end of the document.

```
\newrobustcmd*\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
```

```
\glsignore
```

```
\newcommand*\glsignore}[1]{}
```

2.13 Creating associated files

The `\writeist` command creates the associated customized `.ist` `makeindex` style file. While defining this command, some characters have their catcodes temporarily changed to ensure they get written to the `.ist` file correctly. The `makeindex` actual character (usually `@`) is redefined to be a `?`, to allow internal commands to be written to the glossary file output file.

The special characters are stored in `\@gls@actualchar`, `\@gls@encapchar`, `\@gls@levelchar` and `\@gls@quotechar` to make them easier to use later, but don't change these values, because the characters are encoded in the command definitions that are used to escape the special characters (which means that the user no longer needs to worry about `makeindex` special characters).

The symbols and numbers label for group headings are hardwired into the .ist file as `glsymbols` and `glsnumbers`, the group titles can be translated (so that `\glsymbolsgroupname` replaces `glsymbols` and `\glsnumbersgroupname` replaces `glsnumbers`) using the command `\glsgetgrouptitle` which is defined in `glossary-hypernav`. This is done to prevent any problem characters in `\glsymbolsgroupname` and `\glsnumbersgroupname` from breaking hyperlinks.

`\glsopenbrace` Define `\glsopenbrace` to make it easier to write an opening brace to a file.

```
\edef\glsopenbrace{\expandafter@gobble\string\{}
```

`\glsclosebrace` Define `\glsclosebrace` to make it easier to write an opening brace to a file.

```
\edef\glsclosebrace{\expandafter@gobble\string\}}
```

`\glsbackslash` Define `\glsbackslash` to make it easier to write a backslash to a file.

```
\edef\glsbackslash{\expandafter@gobble\string\}}
```

`\glsquote` Define command that makes it easier to write quote marks to a file in the event that the double quote character has been made active.

```
\edef\glsquote#1{\string"#1\string"}
```

`\glspercentchar` Define `\glspercentchar` to make it easier to write a percent character to a file.

```
\edef\glspercentchar{\expandafter@gobble\string\%}
```

`\glsildechar` Define `\glsildechar` to make it easier to write a tilde character to a file.

```
\edef\glsildechar{\string~}
```

`@glsfirstletter` Define the first letter to come after the digits 0, . . . ,9. Only required for `xindy`.

```
\ifglsxindy
  \newcommand*{\@glsfirstletter}{A}
\fi
```

`SetXdyFirstLetterAfterDigits` Sets the first letter to come after the digits 0, . . . ,9. The starred version sanitizes.

```
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnoxindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
```

`\@xdynumbergrouporder` Specifies the order of the number group.

```

\ifglxindy
  \newcommand*\@xdynumbergrouporder{:before \string"@@glfirstletter\string"}
\fi

```

`\GlsSetXdyNumberGroupOrder` Sets the relative location of the number group. The starred version sanitizes.

```

\newcommand*\GlsSetXdyNumberGroupOrder[1]{%
  \ifstar\s@GlsSetXdyNumberGroupOrder\GlsSetXdyNumberGroupOrder
}
\ifglxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder[1]{%
    \renewcommand*\@xdynumbergrouporder{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder[1]{%
    \renewcommand*\@xdynumbergrouporder{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*\@GlsSetXdyNumberGroupOrder[1]{%
    \glxnoindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*\s@GlsSetXdyNumberGroupOrder{%
    \@GlsSetXdyNumberGroupOrder}
\fi

```

`\@glxminrange` Define the minimum number of successive location references to merge into a range.

```

\newcommand*\@glxminrange{2}

```

`\GlsSetXdyMinRangeLength` Set the minimum range length. The value must either be none or a positive integer. The glossaries package doesn't check if the argument is valid, that is left to xindy.

```

\ifglxindy
  \newcommand*\GlsSetXdyMinRangeLength[1]{%
    \renewcommand*\@glxminrange{#1}}
\else
  \newcommand*\GlsSetXdyMinRangeLength[1]{%
    \glxnoindywarning\GlsSetXdyMinRangeLength}
\fi

```

```

\writeist
\ifglxindy
Code to use if xindy is required.
  \def\writeist{%
Define write register if not already defined
  \ifundef{glswrite}{\newwrite\glswrite}{}%
Update attributes list
  \@gls@addpredefinedattributes

```

Open the file.

```
\openout\glswrite=\istfilename
```

Write header comment at the start of the file

```
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
```

Specify the required styles

```
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite(){(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
```

List the allowed attributes (possible values used by the format key)

```
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
```

Define any additional alphabets

```
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
```

Define location classes.

```
\write\glswrite{^^J; location class definitions^^J}%
```

As from version 3.0, locations are now specified as $\langle Hprefix \rangle \langle number \rangle$, so need to add all possible combinations of location types.

```
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
```

Case where $\langle Hprefix \rangle$ is empty:

```
\protected@write\glswrite(){(define-location-class
\string"\@gls@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
```

Nested iteration over all classes:

```
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite(){(define-location-class
\string"\@gls@classII-\@gls@classI\string"
```


Add all the xindy-only macro definitions (needed to prevent errors in the event that the user changes from xindy to makeindex)

```

\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
      {%
        \string\setentrycounter
          [\expandafter\@gobble\string\#1]{\@this@ctr}%
        \expandafter\string
        \csname\@this@attr\endcsname
          {\expandafter\@gobble\string\#2}%
      }%
    }%
  }%
}%

```

Add the end part of the open tag and the rest of the markup-index information:

```

\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string"\glspercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}}%

```

Specify what to put between letter groups

```

\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%

```

Specify what to put between entries

```

\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%

```

Specify how to format entries

```

\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%

```

Specify how to separate location numbers

```

\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%

```

Specify how to indicate location ranges

```

\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%

```

Specify 2-page and 3-page suffixes, if defined. First, the values must be sanitized to write them explicitly.

```

\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi

```

Specify how to format locations.

```

\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%

```

Specify how to separate letter groups.

```

\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%

```

Define letter group headings.

```

\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%

```

Define additional letter groups.

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%

```

Define additional sort rules

```

\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%

```

Hook for any additional information:

```

\@gls@writeisthook

```

Close the style file

```

\closeout\glswrite

```

Suppress any further calls.

```

\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else

```

Code to use if makeindex is required.

```

\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"%}
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}

\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix

```

```

\string\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
\string"\string}\string\relax
\string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi

```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the file and disable \writeist.

```

\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi

```

`\GlsSetWriteIstHook` Allow user to append information to the style file.

```

\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook

```

`\@gls@writeisthook`

```
\newcommand*\@gls@writeisthook}{}
```

`\GlsSetQuote` Allow user to set the `makeindex` quote character. This is primarily for ngerman users who want to use `makeindex`'s `-g` option.

```

\ifglxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}}

```

If German is in use, set the extra `makeindex` option so `makeglossaries` can pick it up.

```

\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%

```

```
}%
{}%
}%
{}%
```

Need to redefine \@gls@checkquote

```
\edef\@gls@docheckquotedef{%
\noexpand\def\@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@checkesactual###1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkeschar###1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%

\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null

```

```

\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \noexpand\@gls@quotechar\noexpand\string\expandonce{%
    \csname#1\endcsname}\noexpand\@gls@quotechar
  \noexpand\@gls@quotechar\noexpand\string\expandonce{%
    \csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \noexpand\@gls@quotechar\noexpand\string
  \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
  \expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
  \noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
  ###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi

```

\@gls@extramakeindexopts

```
\newcommand*\@gls@extramakeindexopts}[1]{}
```

The command `\noist` will suppress the creation of the `.ist` file. Obviously you need to use this command before `\writeist` to have any effect.

\noist

```
\newcommand{\noist}{%
```

Update attributes list

```
\@gls@addpredefinedattributes
```

```

\let\writeist\relax
}

```

`\@gls@noist@toolate`

```

\newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}

```

`\@makeglossary` is an internal command that takes an argument indicating the glossary type. This command will create the glossary file required by `makeindex` for the given glossary type, using the extension supplied by the `<out-ext>` parameter used in `\newglossary` (and it will also activate the `\glossary` command, and create the customized `.ist` `makeindex` style file).

Note that you can't use `\@makeglossary` for only some of the defined glossaries (with just the base `glossaries` package). You either need to have a `\makeglossaries` for all glossaries or none (otherwise you will end up with a situation where `TEX` is trying to write to a non-existent file). The relevant glossary must be defined prior to using `\@makeglossary`. `glossaries-extra` allows for a hybrid approach.

`\@makeglossary` Unstarred form of `\ifglossaryexists` is used as `\@makeglossary` can't be used with an ignored glossary.

```

\newcommand*{\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%

```

Only create a new write if `savewrites=false` otherwise create a token to collect the information.

```

  \ifglssavewrites
  \expandafter\newtoks\csname glo@#1@filetok\endcsname
  \else
  \expandafter\newwrite\csname glo@#1@file\endcsname
  \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
  \fi
  \@gls@renewglossary
  \writeist
}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘#1’ not defined}%
  {New glossaries must be defined before using \string\makeglossaries}%
}%
}

```

`\@glsopenfile` Open write file associated with the given glossary.

```

\newcommand*{\@glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file

```

```

        \jobname.\csname @glotype@#2@out\endcsname}%
    }

```

\@closegls

```

\newcommand*{\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
}

```

\@gls@automake Unstarred form of \ifglossaryexists is used as \@gls@automake can't be used with an ignored glossary.

```

\ifglsxindy
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-M ord/letorder }}%
        {\let\@gls@order\@empty}%
        \gls@xdy@set@lang{#1}%

        \ifdefvoid\gls@codepage
        {\let\@gls@codepage\@empty}%
        {\def\@gls@codepage{-C \gls@codepage\space}}%
        \@gls@run@unrestricted@shell{xindy
            -I xindy
            \@gls@order
            -L \@gls@langmod\space
            -M \@gls@run@output@file{\gls@istfilebase}
            \@gls@codepage
            -t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
            -o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
            \@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}%
        }%
    }%
    {%
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}

```

\else

```

\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-1 }}%
        {\let\@gls@order\@empty}%
        \@gls@runshell{makeindex \@gls@order
            -s \@gls@run@output@file{\istfilename}
            -t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
            -o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
        }
    }
}

```

```

        \@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}%
    }%
}%
{%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi

```

`\@gls@automake@immediate` Unstarred form of `\ifglossaryexists` is used as `\@gls@automake@immediate` can't be used with an ignored glossary. Note that “out” means the output file from this package's point of view. It will be the input file from the indexing application's point of view.

```

\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%

```

Set `\@gls@langmod`:

```

    \gls@xdy@set@lang{#1}%

    \ifdefvoid\gls@codepage
    {\let\@gls@codepage\@empty}%
    {\def\@gls@codepage{-C \gls@codepage\space}}%
    \@gls@run@unrestricted@shell{xindy
        -I xindy
        \@gls@order
        -L \@gls@langmod\space
        -M \@gls@run@output@file{\gls@istfilebase}
        \@gls@codepage
        -t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
        -o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
        \@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}%
    }%
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
    doesn't exist. Rerun may be required}}%
}%
{%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}

```

```

{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-1 }}%
{\let\@gls@order\@empty}%
\@gls@runshell{makeindex \@gls@order
-s \@gls@run@output@file{\istfilename}
-t \@gls@run@output@file{\jobname.\csuse{@glotype@#1@log}}
-o \@gls@run@output@file{\jobname.\csuse{@glotype@#1@in}}
\@gls@run@output@file{\jobname.\csuse{@glotype@#1@out}}}%
}%
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi

```

`\warn@nomakeglossaries` Issue warning that `\makeglossaries` hasn't been used.

```
\newcommand*{\@warn@nomakeglossaries}{}

```

Only use this if warning if `\printglossary` has been used without `\makeglossaries`

```
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}

```

`\@gls@automake@types` The list of glossaries to build if automake is on. Provided for `glossaries-extra`'s hybrid option.

```
\newcommand{\@gls@automake@types}{\@glo@types}

```

`\@gls@@automake@immediate`

```

\newcommand{\@gls@@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{}%
{%
\IfFileExists{\jobname.aux}%
{\@gls@run@unrestricted@shell{\@gls@automake@makegloss\space \jobname}}{}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
}%

```

```

\fi
}

```

`\makeglossaries` will use `\@makeglossary` for each glossary type that has been defined. New glossaries need to be defined before using `\makeglossary`, so have `\makeglossaries` redefine `\newglossary` to prevent it being used afterwards.

```

\makeglossaries

```

```

\newcommand*{\makeglossaries}{%
\ifglxindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
}%

```

If `automake=immediate` setting is on, use the shell escape now.

```

\@gls@@automake@immediate

```

Define the write used for style file also used for all other output files if `savewrites=true`.

```

\ifundef{\glswrite}{\newwrite\glswrite}{}%

```

If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}

```

If `\@@gls@extramakeindexopts` has been defined, write it:

```

\ifundef\@@gls@extramakeindexopts
{}%
{%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}}
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@@gls@extramakeindexopts}}%
}%

```

Write the name of the style file to the aux file (needed by `makeglossaries`)

```

\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}

```

Iterate through each glossary type and activate it.

```

\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{}}{}{}%
\@makeglossary{\@glo@type}}%
}%

```

New glossaries must be created before `\makeglossaries` so disable `\newglossary`.

```
\renewcommand*\newglossary[4] []{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
```

Any subsequent instances of this command should have no effect. The deprecated `\makeglossary` is not redefined here as it either implements `\makeglossaries` or has been restored to its original definition (in which case it shouldn't be changed).

```
\let\@makeglossary\@gobble
\let\makeglossaries\relax
```

Disable all commands that have no effect after `\makeglossaries`

```
\@disable@onlypremakeg
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Suppress warning about no `\makeglossaries`

```
\let\warn@nomakeglossaries\relax
```

Activate warning about missing `\printglossary`

```
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
```

Declare list parser for `\glsdisplaynumberlist`

```
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}%
\@gls@dodolistparser
\fi
```

Prevent user from also using `\makenoidxglossaries`

```
\let\makenoidxglossaries\@no@makeglossaries
```

Prohibit sort key in `printgloss` family:

```
\renewcommand*\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
```

Check the automake setting:

```
\ifglsautomake
\renewcommand*\@gls@doautomake}{%
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
```

Check the sort setting:

```
\@glo@check@sortallowed\makeglossaries
}%
}
```

Must occur in the preamble:

```
\@onlypreamble{\makeglossaries}
```

`\makeglossaries@warn@noprintglossary`

```
\newcommand\@makeglossaries@warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
```

`\glswrite` The definition of `\glswrite` has now been moved to `\makeglossaries` so that it's only defined if needed.

If `\makeglossaries` hasn't been used, issue a warning. Also issue a warning if neither `\printglossaries` nor `\printglossary` have been used.

```
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
```

`\makenoidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```
\newcommand*\makenoidxglossaries{%
\def\glsindexingsetting{noidx}%
\@domakeglossaries
}%
```

Redefine empty glossary warning:

```
\renewcommand{\@gls@noref@warn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
```

Don't escape makeindex/xindy characters:

```
\let\@gls@checkmkidxchars\@gobble
```

Don't escape locations:

```
\glseclocationsfalse
```

Write glossary information to aux instead of glossary files

```
\let\@do@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
\renewcommand{\@do@seeglossary}[2]{%
\protected@edef\@gls@label{\glsdetoklabel{##1}}%
\protected@write\@auxout{}{%
\string\@gls@reference
  {\csname glo@\@gls@label @type\endcsname}%
  {\@gls@label}%
  {%
  \string\glsseeformat##2}%
  }%
}%
```

If user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\AtBeginDocument
{%
\write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
```

Change warning about no glossaries

```
\def\warn@noprintglossary{%
\GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%
```

Suppress warning about no \makeglossaries

```
\let\warn@nomakeglossaries\relax
```

Prevent user from also using \makeglossaries

```
\let\makeglossaries\@no@makeglossaries
```

Allow sort key in printgloss family:

```
\renewcommand*\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@assign@sortkey
```

Initialise default sort order:

```
\def\@glo@sorttype{\@glo@default@sorttype}%
}%
```

All entries must be defined in the preamble:

```
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoid@glossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
```

Redefine `\glsentrynumberlist`

```
\renewcommand*\glsentrynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoid@loclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}%
```

Redefine `\glsdisplaynumberlist`

```
\renewcommand*\glsdisplaynumberlist[1]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noid@loclist@sep{%
      \def\@gls@noid@loclist@sep{%
        \def\@gls@noid@loclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noid@loclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noid@loclist@finalsep{}%
  \def\@gls@noid@loclist@prev{}%
  \forlistloop{\glsnoid@display@loclist@handler}{\@gls@loclist}%
  \@gls@noid@loclist@finalsep
  \@gls@noid@loclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
```

```
}%
}%
```

Provide a generic way of iterating through the number list:

```
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
```

Modify sanitize sort function

```
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
```

Check sort option allowed.

```
\@glo@check@sortallowed\makenoidxglossaries
```

Implement nonumberlist

```
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
```

Preamble-only command:

```
\@onlypreamble{\makenoidxglossaries}
```

```
\glsnumberlistloop{<label>}{<handler>}{<xr handler cs>}
```

\glsnumberlistloop

```
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
```

`\noidxnumberlistloophandler` Handler macro for `\glsnumberlistloop`. (The argument should be in the form `\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<n>}`)

```
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
```

`\@no@makeglossaries` Can't use both `\makeglossaries` and `\makenoidxglossaries`

```
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
```

`\@gls@noref@warn` Warning when no instances of `\@gls@reference` found.

```
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
```

2.14 Writing information to associated files

`\gls@noidxglossary` Write the glossary information to the aux file (for the 'noidx' method).

```
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@glslocref}}%
    }%
  }%
}
```

`\istfile` Deprecated in v3.0 (2011-04-02). Removed in v4.50.

At the end of the document, the files should be created if `savewrites=true`.

```
\AtEndDocument{%
  \glswritefiles
}
```

`\@glswritefiles` Only write the files if `savewrites=true`.

```
\newcommand*{\@glswritefiles}{%
  Iterate through all the glossaries.
  \forallglossaries{\@glo@type}{%
```

Check for empty glossaries (patch provided by Patrick Häcker)

```

\ifcsundef{glo@\@glo@type @filetok}%
{%
  \def\gls@tmp{}%
}%
{%
  \edef\gls@tmp{\expandafter\the
    \csname glo@\@glo@type @filetok\endcsname}%
}%
\ifx\gls@tmp\@empty
  \ifx\@glo@type\glsdefaulttype
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries.^^JRemember to use package option ‘nomain’ if
you
      don’t want to^^Juse the main glossary}%
  \else
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries}%
  \fi
\else
  \@glsopenfile{\glswrite}{\@glo@type}%
  \immediate\write\glswrite{%
    \expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
  \immediate\closeout\glswrite
  \fi
}%
}

```

As from v4.10, the `\glossary` command isn’t used by the `glossaries` package. Since the user isn’t expected to use this command (as `glossaries` takes care of the particular format required for `makeindex/xindy`) there’s no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```

\gls@glossary
  \newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
  }

```

```

\@gls@glossary{<type>}{<indexing info>}

```

`\@gls@glossary`

(In v4.10, `\@glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within `<indexing info>` (so you can do, for example, `\index{%\%}`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the `sort key`), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glsperscentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```
\newcommand*{\@gls@glossary}[2]{%
  \ifgls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}
```

This is a convenience command to set `\@gls@glossary`. It's used by `\makeglossary` and then redefined to do nothing, as it only needs to be done once.

`\@gls@renewglossary`

```
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begingroup\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
```

The `\gls@wrglossary` command is defined to have two arguments. The first argument is the glossary type, the second argument is the glossary entry (the format of which is set in `\glslink`).

`\gls@wrglossary`

```
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
      {%
        \expandafter\protected@write\csname glo@#1@file\endcsname{%
          \gls@disablepagerefexpansion}{#2}%
        }
    \else
      \csname glo@#1@file\endcsname
    \fi
  }
}
```

```

    }%
    {%
    \ifignoredglossary{#1}{}%
    {%
    \GlossariesWarning{No file defined for glossary ‘#1’}%
    }%
    }%
  \fi
  \endgroup\@esphack
}

```

`\do@wrglossary`

```

\newcommand*\do@wrglossary[1]{%
  \gls@ifnotmeasuring{\glswriteentry{#1}{\do@wrglossary{#1}}}%
}

```

`\glswriteentry` Provide a user level command so the user can customize whether or not a line should be added to the glossary. The arguments are the label and the code that writes to the glossary file.

```

\newcommand*\glswriteentry[2]{%
  \ifglsindexonlyfirst
  \ifglsused{#1}{#2}%
  \else
  #2%
  \fi
}

```

`\gls@protected@pagefmts` List of page formats to be protected against expansion.

```

\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
  \gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

```

`\gls@disablepagerefexpansion` As from v4.50, this will only do the page protection formats for `esclocations=true`.

```

\newcommand*\gls@disablepagerefexpansion{%
  \glswrglossdisablelocationcmds
  \ifglsesclocations
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
  \expandafter\let\@gls@this\relax
  }%
  \fi
}

```

`\glswrglossdisablelocationcmds`

```

\newcommand{\glswrglossdisablelocationcmds}{}

```

`\gls@alphpage`

```

\newcommand*\gls@alphpage{\@alph\c@page}

```

```

\gls@Alphpage
\newcommand*\gls@Alphpage{\@Alph\c@page}

\gls@numberpage
\newcommand*\gls@numberpage{\number\c@page}

\gls@arabicpage
\newcommand*\gls@arabicpage{\@arabic\c@page}

\gls@romanpage
\newcommand*\gls@romanpage{\romannumeral\c@page}

\gls@Romanpage
\newcommand*\gls@Romanpage{\@Roman\c@page}

```

```
\glsaddprotectedpagefmt{<cs name>}
```

```
\glsaddprotectedpagefmt
```

Added a page format to the list of protected page formats. The argument should be the name (without a backslash) of the command that takes a \TeX register as the argument ($\langle csname \rangle \c@page$ must be valid).

```

\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifgllesclocations
    \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@gl#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gl#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}

```

```
\GlsWarnAddProtectedPageFmt
```

```

\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}

```

```
\@wrglossarynumberhook Hook used by \@do@wrglossary
```

```
\newcommand*\@wrglossarynumberhook{}
```

`\@wrglossary@pageformat`

```
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
```

`\@do@wrglossary` Write the glossary entry in the appropriate format.

```
\newcommand*{\@do@wrglossary}[1]{%
  \ifglseclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
```

`glswrglossdisableanchorcmds`

```
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
}
```

`\glswrglosslocationtarget` Must expand. May be used to alter the location suffix in the `\glshypernumber` anchor.

```
\newcommand{\glswrglosslocationtarget}[1]{#1}
```

`\glswrglosslocationtextfmt` Used to encapsulate the hyperlink text in `\glshypernumber`.

```
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
```

`\@do@noesc@wrglossary` Write the glossary entry in the appropriate format. The locations don't need to be pre-processed before writing the information to the glossary file, but the prefix still needs to be found.

```
\newcommand*{\@do@noesc@wrglossary}[1]{%
```

Don't fully expand yet.

```
\expandafter\def\expandafter@glslcref\expandafter{\theglslentrycounter}%
\expandafter\def\expandafter@glshlcref\expandafter{\theHglslentrycounter}%
```

Find the prefix if `\@glshlcref` and `\@glslcref` aren't the same.

```
\ifx\@glshlcref\@glslcref
  \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 `\theglslentrycounter` and `\theHglslentrycounter` at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

```
\@xp@glsl@getcounterprefix{\@glslcref}{\@glshlcref}%
\fi
```

De-tok label if required.

```
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
```

Write the information to file:

```
\@@do@wrglossary  
}
```

`\ifglswrallowprimitivemods` Conditional to determine whether or not `\@@do@esc@wrglossary` should be allowed to temporarily redefine `\the` and `\number`. Version 4.50 switched the default to false.

```
\newif\ifglswrallowprimitivemods  
\glswrallowprimitivemodsfalse
```

`\@@do@esc@wrglossary` Write the glossary entry in the appropriate format. (Need to set `\@glsnumberformat` and `\@gls@counter` prior to use.) The argument is the entry's label. This is far more complicated with `xindy` than with other indexing methods. There are two necessary but conflicting requirements with `xindy`:

1. all backslashes in the location must be escaped;
2. `\c@page` can't be prematurely expanded.

(With `makeindex` there's the remote possibility that the page compositor is a `makeindex` special character, so that would also need to be escaped.)

For example, suppose `\thepage` is defined as

```
%\renewcommand{\thepage}{\tally{page}}  
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}  
%
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won't accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character "t". The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn't be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn't be escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}  
%
```

(note the use of `\expandafter` and `\name`) or in the form

```
%\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}  
%
```

In the second case, `\arabic` is one of the known commands that's temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it's followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH<counter>` is defined in the form `<prefix>.\the<counter>`. It's possible (although again unlikely) that a `makeindex` character might occur in the prefix, which therefore needs escaping. The prefix is passed as the optional argument of `\setentrycounter` which is needed by commands like `\glshypernumber` to create a hyperlink for a given counter (like `\hyperpage` but for an arbitrary counter).

```
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!  
  \begingroup
```

First a bit of hackery to prevent premature expansion of `\c@page`. Store original definitions (scoped):

```
\let\gls@orgthe\the  
\let\gls@orgnumber\number  
\let\gls@orgarabic\@arabic  
\let\gls@orgromannumeral\romannumeral  
\let\gls@orgalph\@alph  
\let\gls@orgAlph\@Alph  
\let\gls@orgRoman\@Roman
```

Redefine:

```
\ifglswrallowprimitivemods
```

The redefinition of `\the` to use `\expandafter` solves the problem of `\the\csname c@<counter>\endcsname` but is only a partial solution to the problem of `\the\value`. With `\value`, `\c@page` is too deeply hidden and will be expanded too soon, but at least there won't be an error.

```
\def\gls@the##1{%  
  \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%  
\def\the{\expandafter\gls@the}%  
\def\gls@number##1{%  
  \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%  
\def\number{\expandafter\gls@number}%
```

```

\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
  \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%

```

Add hook to allow for other number formats:

```
\@wrglossarynumberhook
```

Prevent expansion:

```
\gls@disablepagerefexpansion
```

Now store location in \@glslocref:

```

\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup

```

Escape any special characters. It's possible that with `makeindex` the separator might be a `makeindex` special character. Although not likely, it still needs to be taken into account.

```
\@gls@checkmkidxchars\@glslocref
```

Check if the hyper-location is the same as the location and set the hyper prefix.

```

\expandafter\ifx\theHglentrycounter\theglsentrycounter\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@mkidxrangear\@glsnumberformat\@nil
\def\@glo@range{}%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi

```

Write to the glossary file using xindy syntax.

```

\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else

```

Convert the format information into the format required for makeindex

```

\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
{\@glo@counterprefix}%

```

Write to the glossary file using makeindex syntax.

```

\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string@glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}

```

\@xp@gls@getcounterprefix

```

\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgrou
\gls@wrglossdisableanchorcmds
\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
      :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
}

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{#{#2#3}}%
  \fi
}
```

`\@@gls@fixbraces`

```
\def\@@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
```

```
\glssee \glssee{<label>}{<cross-ref list>}
\newrobustcmd*{\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*{\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
```

`\glsseeformat` The first argument specifies what tag to use (e.g. “see”), the second argument is a comma-separated list of labels. The final argument (the location) is ignored.

```
\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
```

`\glsseelist` `\glsseelist{<list>}` formats list of entry labels.

```
\newrobustcmd*{\glsseelist}[1]{%
```

If there is only one item in the list, set the last separator to do nothing.

```
\let\@gls@dolast\relax
```

Don’t display separator on the first iteration of the loop

```
\let\@gls@donext\relax
```

Iterate through the labels

```
\@for\@gls@thislabel:=#1\do{%
```

Check if on last iteration of loop

```
\ifx\@xfor@nextelement\@nnil
```

```
\@gls@dolast
```

```
\else
```

```
\@gls@donext
```

```
\fi
```

Display the entry for this label. (Expanding label as it’s a temporary control sequence that’s used elsewhere.)

```
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
```

Update separators

```
\let\@gls@dolast\glsseelastsep
```

```
\let\@gls@donext\glsseesep
```

```
}%
```

```
}
```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```
\newcommand*\glsseelastsep{\space\andname\space}
```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```
\newcommand*\glsseesep}{, }
```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```
\newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
```

`\glsseeitemformat` Introduced in v3.0 to replace use of `\glsentryname` (originally in `\glsseeitem`) with `\glsentrytext` to avoid problems with the name key being sanitized. This ceased being a problem in v3.08a, but `\glsentrytext` was retained for backward-compatibility. As from v4.50, the default has reverted back to using `\glsentryname` unless the entry is an acronym (which could have a potentially long name).

```
\newcommand*\glsseeitemformat[1]{%
  \ifglshashshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
```

2.16 Displaying the glossary

An individual glossary is displayed in the text using `\printglossary[<key-val list>]`. If the `type` key is omitted, the default glossary is displayed. The optional argument can be used to specify an alternative glossary, and can also be used to set the style, title and entry in the table of contents. Available keys are defined below.

`\gls@save@numberlist` Provide command to store number list.

```
\newcommand*\gls@save@numberlist[1]{%
  \ifglssavenumberlist
    \toks@{#1}%

    \protected@edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\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=glsdefaulttype]{%
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\@printglossary{#1}{\@print@glossary}%
}
```

The `\printglossaries` command will do `\printglossary` for each glossary type that has been defined. It is better to use `\printglossaries` rather than individual `\printglossary` commands to ensure that you don't forget any new glossaries you may have created. It also makes it easier to chop and change the value of the acronym package option. However, if you want to list the glossaries

in a different order, or if you want to set the title or table of contents entry, or if you want to use different glossary styles for each glossary, you will need to use `\printglossary` explicitly for each glossary type.

```

\printglossaries
  \newcommand*\printglossaries}{%
    \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
  }

\printnoidxglossary Provide an alternative to \printglossary that doesn't require an external indexing application.
  \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 @title\endcsname}%

  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '@@glo@type'. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%

```

Store current value of `\glossaryentrynumbers`. (This may be changed via the optional argument)

```
\let\org@glossaryentrynumbers\glossaryentrynumbers
```

Localise the effects of the optional argument

```
\gls@printglossary@begin
```

Activate or deactivate sort key:

```
\@printgloss@setsort
```

Determine settings specified in the optional argument.

```
\setkeys{printgloss}{#1}%
```

Does the glossary exist?

```
\@printgloss@checkexists{\@glo@type}%  
{%
```

If title has been set, but `toctitle` hasn't, make `toctitle` the same as given title (rather than the title used when the glossary was defined)

```
\ifx\glossarytitle\org@glossarytitle  
\else  
\expandafter\let\csname @glo@type @title\endcsname  
\glossarytitle
```

```
\fi
```

Allow a high-level user command to indicate the current glossary

```
\let\currentglossary\@glo@type
```

Enable individual number lists to be suppressed.

```
\let\org@glossaryentrynumbers\glossaryentrynumbers  
\let\glsnonextpages\@glsnonextpages
```

Enable individual number list to be activated:

```
\let\glsnextpages\@glsnextpages
```

Enable suppression of description terminators.

```
\let\nopostdesc\@nopostdesc
```

Set up the entry for the TOC

```
\gls@dotoc@title
```

Set the glossary style

```
\@glossarystyle
```

Added a way to fetch the current entry label (v3.08 updated for new `\glossentry` and `\subglossentry`, but this is now only needed for backward compatibility):

```
\let\gls@org@glossaryentryfield\glossentry  
\let\gls@org@glossarysubentryfield\subglossentry  
\renewcommand{\glossentry}[1]{%  
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%  
\gls@org@glossaryentryfield{##1}%  
}%  
\renewcommand{\subglossentry}[2]{%
```

```

\protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%

```

```
\gls@preglossaryhook
```

Now do the handler macro that deals with the actual glossary:

```

#2%
}%

```

End the current scope

```
\gls@printglossary@end
```

Reset `\glossaryentrynumbers`

```
\global\let\glossaryentrynumbers\org@glossaryentrynumbers
```

Suppress warning about no `\printglossary`

```

\global\let\warn@noprntglossary\relax
}

```

```
\gls@printglossary@begin
```

```
\newcommand{\gls@printglossary@begin}{\begingroup}
```

`\gls@printglossary@end` See <https://github.com/latex3/tagging-project/issues/48>

```

\newcommand{\gls@printglossary@end}{%
\expandafter\endgroup\if@endpe\@doendpe\fi
}

```

`\@print@glossary` Internal workings of `\printglossary` dealing with reading the external file.

```
\newcommand{\@print@glossary}{%
```

Some macros may end up being expanded into internals in the glossary, so need to make `@` a letter. (Unlikely to be a problem since v3.08a but kept for backward compatibility.)

```
\makeatletter
```

Input the glossary file, if it exists.

```
\input@{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
```

If the glossary file doesn't exist, do `\null`. (This ensures that the page is shipped out and all write commands are done.) This might produce an empty page, but at this point the document isn't complete, so it shouldn't matter.

```

\IfFileExists{\jobname.\csname @glo@type@\glo@type @in\endcsname}%
{}%
{\null}%

```

If `xindy` is being used, need to write the language dependent information to the `.aux` file for `makeglossaries`.

```
\ifglxindy
```

Set `\@gls@langmod`:

```

\gls@xdy@set@lang{\glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%

```

If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\noexpand\immediate\noexpand\write\@auxout{%
  \string\providecommand\string\@xdylanguage[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%
  \string\@xdylanguage{\@glo@type}{\@gls@langmod}}%
}%
}%
\do@auxoutstuff
\edef\@do@auxoutstuff{%
  \noexpand\AtEndDocument{%

```

If the user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\noexpand\immediate\noexpand\write\@auxout{%
  \string\providecommand\string\@gls@codepage[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%
  \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
}%
}%
\do@auxoutstuff
\fi

```

Activate warning if `\makeglossaries` hasn't been used.

```

\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
  hasn't been used,^^Jthe glossaries will not be updated}%
}%
}

```

The sort macros all have the syntax:

```

\@glo@sortmacro@<order>{<type>}

```

where *<order>* is the sort order as specified by the sort key and *<type>* is the glossary type. (The referenced entry list is stored in `\@glsref@<type>`). The actual sorting is done by `\@glo@sortentries{<handler>}{<type>}`.

New to v4.55: Check if `datatool-base v3.0+` is installed, which provides the command `\datatool_sortwordseq:NN`

```
\ExplSyntaxOn
```

Scratch variables.

```

\tl_new:N \l__glossaries_sort_value_tl
\tl_new:N \l__glossaries_entry_label_tl
\tl_new:N \l__glossaries_entry_letter_title_tl
\tl_new:N \l__glossaries_entry_letter_group_tl

```

Check if new datatool sort function available:

```
\cs_if_exist:NTF \datatool_sortwordseq:NN  
{
```

Sequence variable in which to store entries that need sorting.

```
\seq_new:N \l__glossaries_actual_sort_seq
```

Actual sequence uses markup to expand to either the label or the provided sort value. This is to assist datatool-base localisation group handlers.

```
\cs_new:Nn \__glossaries_actual_sort:nn  
{  
  \exp_not:n { #1 }  
}  
\cs_new:Nn \__glossaries_actual_label:nn  
{  
  \exp_not:n { #2 }  
}  
\cs_new:Nn \__glossaries_actual:nn  
{  
  \__glossaries_actual_sort:nn { #1 } { #2 }  
}
```

Sequence variable in which to store labels of entries that need sorting.

```
\seq_new:N \l__glossaries_sort_seq
```

Mapping from label to sort value.

```
\prop_new:N \l__glossaries_sort_prop
```

Information message:

```
\msg_new:nnn { glossaries } { presort }  
{  
  Preparing ~ to ~ sort ~ glossary ~ '#1' ~  
  (total ~ number ~ of ~ referenced ~ entries: ~ #2 )  
}
```

Set the sequence from the glossary list.

```
\cs_new:Nn \__glossaries_set_sort_seq:n  
{  
  \msg_note:nnee { glossaries } { presort }  
  { #1 }  
  { \__glossaries_glsref_count:n { #1 } }  
  \seq_clear:N \l__glossaries_sort_seq  
  \seq_clear:N \l__glossaries_actual_sort_seq  
  \prop_clear:N \l__glossaries_sort_prop  
  \forlistcsloop  
  \__glossaries_add_sort_item:n  
  { @glsref@ #1 }  
}
```

Add label to the sequence if not already added.

```
\cs_new:Nn \__glossaries_add_sort_item:n  
{
```

```

\glsdoifexists { #1 }
{
  \seq_if_in:NnF \l__glossaries_sort_seq { #1 }
  {
    \ifglshasparent { #1 }
    {

```

If this entry has a parent, ensure that the parent is also added. This uses recursion to ensure that all ancestors are added.

```

      \exp_args:Nv \__glossaries_add_sort_item:n
      { glo@ #1 @parent }

```

The parent's hierarchical sort value should now be stored in the property list.

```

      \prop_get:NvN \l__glossaries_sort_prop
      { glo@ #1 @parent }
      \l__glossaries_sort_value_tl
    \quark_if_no_value:NTF
      \l__glossaries_sort_value_tl
    {

```

Shouldn't happen, but if it does then assume no parent.

```

      \tl_clear:N \l__glossaries_sort_value_tl
    }
  {

```

Append marker to ensure that child entries are placed immediately after parent entry.

```

      \tl_put_right:Nn \l__glossaries_sort_value_tl
      { \datatoolctrlboundary \datatoolasciistart }
    }
  }
  {

```

No parent, so no hierarchical sort prefix.

```

      \tl_clear:N \l__glossaries_sort_value_tl
    }

```

Append the entry's sort value to ensure hierarchy preserved, if applicable.

```

      \tl_put_right:Nv \l__glossaries_sort_value_tl
      { glo@ #1 @sort }

```

Add the hierarchical sort value to the property list.

```

      \prop_put:NnV \l__glossaries_sort_prop { #1 }
      \l__glossaries_sort_value_tl

```

Add this entry's label to the sequence.

```

      \seq_put_right:Nn \l__glossaries_sort_seq { #1 }

```

Add both the sort value and label to the actual sequence.

```

      \seq_put_right:Ne \l__glossaries_actual_sort_seq
      {
        \exp_not:N \__glossaries_actual:nn
        { \exp_not:V \l__glossaries_sort_value_tl }
      }

```

```

        { #1 }
      }
    }
  }
}

```

Fetch the sort value from the property map:

```

\cs_new:Nn \__glossaries_fetch_sort:n
{
  \group_begin:
    \cs_set_eq:NN \__glossaries_actual:nn \__glossaries_actual_label:nn
    \exp_args:NNNe
  \group_end:
    \tl_set:Nn \l__glossaries_entry_label_tl { #1 }
  \prop_get:NVN \l__glossaries_sort_prop
    \l__glossaries_entry_label_tl
    \l__glossaries_sort_value_tl
}

```

Available sort methods. Word sort:

```

\cs_new:Nn \__glossaries_sort_word:n
{
  \__glossaries_do_sort:Nn
    \__glossaries_sort_word_handler:nN
    { #1 }
}
\cs_new:Nn \__glossaries_sort_word_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \exp_args:NV \DTLsortwordhandler
    \l__glossaries_sort_value_tl #2
}

```

Letter sort:

```

\cs_new:Nn \__glossaries_sort_letter:n
{
  \__glossaries_do_sort:Nn
    \__glossaries_sort_letter_handler:nN
    { #1 }
}
\cs_new:Nn \__glossaries_sort_letter_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \exp_args:NV \DTLsortletterhandler
    \l__glossaries_sort_value_tl #2
}

```

Case-sensitive.

```

\cs_new:Nn \__glossaries_sort_case:n
{
  \__glossaries_do_sort:Nn

```

```

    \__glossaries_sort_case_handler:nN
    { #1 }
  }

```

Closest match for original case-sensitive sorting. This omits the localisation support.

```

\cs_new:Nn \__glossaries_sort_case_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }

```

This preprocessing may have already been done if the sort value was obtained by expanding the name, but this may not have happened (e.g. the sort value was explicitly set).

```

  \bool_set_false:N \l_datatool_sort_to_lowercase_bool
  \exp_args:NNV \datatool_sort_preprocess:Nn
    #2 \l__glossaries_sort_value_tl
  \exp_args:NV \DTLsortwordcasehandler
    \l__glossaries_sort_value_tl #2
  \tl_set:Ne #2 { \text_purify:n { #2 } \datatoolctrlboundary }
}

```

Case-insensitive.

```

\cs_new:Nn \__glossaries_sort_nocase:n
{
  \__glossaries_do_sort:Nn
    \__glossaries_sort_nocase_handler:nN
    { #1 }
}

```

Closest match for original case-insensitive sorting. This omits the localisation support.

```

\cs_new:Nn \__glossaries_sort_nocase_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }

```

Again, this preprocessing may have already been done.

```

  \bool_set_true:N \l_datatool_sort_to_lowercase_bool
  \exp_args:NNV \datatool_sort_preprocess:Nn
    #2 \l__glossaries_sort_value_tl
  \exp_args:NV \DTLsortwordhandler
    \l__glossaries_sort_value_tl #2
  \tl_set:Ne #2 { \text_purify:n { #2 } \datatoolctrlboundary }
}

```

Syntax: $\langle handler-cs \rangle \{ \langle type \rangle \}$

```

\cs_new:Nn \__glossaries_do_sort:Nn
{
  \renewcommand \dtlnonlettergroup [ 1 ] { glssymbols }
  \renewcommand \dtlnumbergroup [ 1 ] { glsnumbers }
  \__glossaries_set_sort_seq:n { #2 }
  \datatool_sortwordseq:NN
    \l__glossaries_actual_sort_seq

```

```

#1
\tl_clear:c { @glsref@ #2 }
\seq_map_inline:Nn \l__glossaries_actual_sort_seq
{
  \tl_set:Ne \l__glossaries_entry_letter_title_tl
  { \exp_args:No \exp_not:o { \DTLsortedletter { ##1 } } }
  \tl_set:Ne \l__glossaries_entry_letter_group_tl
  { \DTLsortedletter { ##1 } }
  \tl_set:Ne \l__glossaries_entry_label_tl { ##1 }
  \group_begin:
  \cs_set_eq:NN \__glossaries_actual:nm \__glossaries_actual_label:nm
  \exp_args:NNNe
  \group_end:
  \tl_set:Nn \l__glossaries_entry_label_tl
  { \l__glossaries_entry_label_tl }
  \exp_args:NnV \listcseadd { @glsref@ #2 } \l__glossaries_entry_label_tl

```

Save letter group information:

```

\tl_gset:ce
{ glo@ \l__glossaries_entry_label_tl @dtlsortgroup }
{
  { \l__glossaries_entry_letter_group_tl }
  { \exp_not:V \l__glossaries_entry_letter_title_tl }
}
}
}

```

Fetch the letter group from the sort value:

```

\cs_new:Npn \@gls@fetch@lettergrp
{
  \tl_if_head_eq_meaning:VNTF \@gls@sort \__datatool_sorted:nm
  {
    \tl_set:Nx \@glo@thislettergrp
    { \exp_args:NV \DTLsortedletter \@gls@sort }
  }
  {
    \exp_args:NVV \DTLassignlettergroup \@gls@sort \@gls@sort
    \@glo@thislettergrp
  }
}
}
{

```

Old version of datatool-base, so use old method.

```

\cs_new:Nn \__glossaries_sort_word:n
{
  \@glo@sortentries{\@glo@sorthandler@word}{#1}
}
\cs_new:Nn \__glossaries_sort_letter:n
{
  \@glo@sortentries{\@glo@sorthandler@letter}{#1}
}

```

```

}
\cs_new:Nn \__glossaries_sort_case:n
{
  \@glo@sortentries{\@glo@sorthandler@case}{#1}
}
\cs_new:Nn \__glossaries_sort_nocase:n
{
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}
}
\cs_new:Npn \@gls@fetch@lettergrp
{
  \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
}
}

```

Provide command that can be used once L3 syntax switched off to ease conversion. (I don't currently have time to rewrite the entire package.)

```

\cs_new:Npn \@@glo@sortentries #1 #2
{
  \cs_if_exist:cTF { __glossaries_sort_ #1 :n }
  {
    \use:c { __glossaries_sort_ #1 :n } { #2 }
  }
  {
    \PackageError { glossaries }
    { Unknown ~ sort ~ method ~ '#1' } { }
  }
}
\ExplSyntaxOff

```

`\@glo@sortentries` Syntax: `\handler-cs{<type>}`

```

\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%

  \protected@edef\@glo@type{#2}%
  \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
  \csdef{\@glsref@#2}{}%
  \@for\@this@label:=\@glo@sortinglist\do{%

```

Has this entry already been added?

```

  \xifinlistcs{\@this@label}{\@glsref@#2}%
  {}%
  {%
    \listcsxadd{\@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{\@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%

```

```

    }%
    {}%
  }%
}

```

```
\@glo@addchildren{<type>}{<parent>}
```

\@glo@addchildren

```
\newcommand*{\@glo@addchildren}[2]{%
```

Scope to allow nesting.

```

  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%

```

Check this label hasn't already been added.

```

    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%

```

Does this child have children?

```

    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    {}%
    }%
  }%

```

```
\egroup
}
```

\@glo@do@sortentries

```
\newcommand*{\@glo@do@sortentries}[1]{%
\ifglshasparent{#1}%
{%

```

This entry has a parent, so add it to the child list

```

  \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
  \ifcsundef{@glo@sortingchildren@\@glo@parent}%
  {%
    \csdef{@glo@sortingchildren@\@glo@parent}{}%
  }%
  {}%
  \expandafter\@glo@sortedinsert
  \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%

```

Has the parent been added?

```

  \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
  {%

```

Yes, it has so do nothing.

```
}%
{%
```

No, it hasn't so add it now.

```
\expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}%
}%
{%
```

`\@glo@sortedinsert{\@glo@sortinglist}{#1}%`

```
}%
}
```

`\@glo@sortedinsert{<list>}{<entry label>}`

`\@glo@sortedinsert`

Insert into list.

```
\newcommand*\@glo@sortedinsert}[2]{%
\dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
```

The sort handlers need to be in the form required by datatool's `\dtl@sortlist` macro. These must set the count register `\dtl@sortresult` to either -1 ($\#1$ less than $\#2$), 0 ($\#1 = \#2$) or $+1$ ($\#1$ greater than $\#2$).

`\@glo@sorthandler@word`

```
\newcommand*\@glo@sorthandler@word}[2]{%
\letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
\edef\glo@do@compare{%
\noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
{\expandonce\@gls@sort@B}%
{\expandonce\@gls@sort@A}%
}%
\glo@do@compare
}
```

`\@glo@sorthandler@letter`

```
\newcommand*\@glo@sorthandler@letter}[2]{%
\letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
\edef\glo@do@compare{%
\noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
{\expandonce\@gls@sort@B}%
{\expandonce\@gls@sort@A}%
}%
\glo@do@compare
}
```

`\@glo@sorthandler@case` Case-sensitive sort.

```
\newcommand*\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{\@glo\@glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{\@glo\@glsdetoklabel{#2}@sort}%
  \edef\@glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \@glo@do@compare
}
```

`\@glo@sorthandler@nocase` Case-insensitive sort.

```
\newcommand*\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{\@glo\@glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{\@glo\@glsdetoklabel{#2}@sort}%
  \edef\@glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \@glo@do@compare
}
```

`\@glo@sortmacro@word` Sort macro for ‘word’

```
\newcommand*\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@@sortentries{word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=word]}{ }%
  }%
}
```

`\@glo@sortmacro@letter` Sort macro for ‘letter’

```
\newcommand*\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@@sortentries{letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=letter]}{ }%
  }%
}
```

`\glo@sortmacro@standard` Sort macro for ‘standard’. (Use either ‘word’ or ‘letter’ order.)

```
\newcommand*\glo@sortmacro@standard}[1]{%
  \ifdefstring{\glo@default@sorttype}{standard}%
  {%
    \glo@sortentries{glsorder}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}%
  }%
}
```

`\glo@sortmacro@case` Sort macro for ‘case’

```
\newcommand*\glo@sortmacro@case}[1]{%
  \ifdefstring{\glo@default@sorttype}{standard}%
  {%
    \glo@sortentries{case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}%
  }%
}
```

`\glo@sortmacro@nocase` Sort macro for ‘nocase’

```
\newcommand*\glo@sortmacro@nocase}[1]{%
  \ifdefstring{\glo@default@sorttype}{standard}%
  {%
    \glo@sortentries{nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=nocase]}{}%
  }%
}
```

`\glo@sortmacro@def` Sort macro for ‘def’. The order of definition is given in `\glo@list@{type}`.

```
\newcommand*\glo@sortmacro@def}[1]{%
  \def\glo@sortinglist{}%
  \forglseentries[#1]{\gls@thislabel}%
  {%
    \xifinlistcs{\gls@thislabel}{\glsref@#1}%
    {%
      \listead{\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.

Version 4.57: switch to constructing token list (as per `\printunsrtglossary`) and check for `dtl-sortgroup` field for letter groups (which will have the format `{\grplabel}{\grouptitle}` if new `datatool` sorting functions available). If this causes any compatibility issues, the original behaviour can be restored by redefining `\@print@noidx@glossary` to use `\@old@print@noidx@glossary` instead.

```
\ExplSyntaxOn  
\newcommand*{\@print@noidx@glossary}{%  
\__glossaries_print_noidx:  
}
```

New function:

```
\tl_new:N \__glossaries_content_tl  
\tl_new:N \__glossaries_entry_prev_letter_group_tl  
\cs_new:Nn \__glossaries_print_noidx:  
{  
\tl_clear:N \__glossaries_content_tl  
\cs_if_exist:cTF { @glsref@ \@glo@type }  
{
```

Sort the entries:

```
\cs_if_exist:cTF {@glo@sortmacro@ \@glo@sorttype}
{
  \use:c { @glo@sortmacro@ \@glo@sorttype } { \@glo@type }
}
{
  \PackageError{ glossaries }
    { Unknown ~ sort ~ handler ~ '\@glo@sorttype' } { }
}
\tl_clear:N \l__glossaries_entry_letter_group_tl
\tl_clear:N \l__glossaries_entry_prev_letter_group_tl
```

Add the glossary heading and preamble

```
\tl_put_right:Nn \l__glossaries_content_tl
{
  \glossarysection [ \glossarytoctitle ] { \glossarytitle }
  \glossarypreamble
  \begin { theglossary }
  \glossaryheader
  \glsresetentrylist
}
}
```

Iterate through the entries.

```
\forlistcsloop \__glossaries_noidx_do:n { @glsref@ \@glo@type }
```

Add final code to end the glossary and do the postamble:

```
\tl_put_right:Nn \l__glossaries_content_tl
{
  \end { theglossary }
  \glossarypostamble
}
}
```

Now do the glossary. This has a hook before it mainly for debugging purposes. For example, redefine the hook to `\show` in order to show the contents to check it's correct.

```
\glsnoidxprecontenthook
\l__glossaries_content_tl
}
{
  \@gls@noref@warn { \@glo@type }
}
}
%\end{macro}
```

```
\glsnoidxprecontenthook
```

```
\newcommand\glsnoidxprecontenthook{}
```

New handler macro for appending content:

```
\cs_new:Nn \__glossaries_noidx_do:n
{
  \tl_if_exist:cTF {glo@\glsdetoklabel{#1}@loclist}
```

```

{
  \tl_set_eq:Nc \@gls@loclist { glo@ #1 @ loclist }
}
{
  \tl_clear:N \@gls@loclist
}
\ifglshasparent { #1 }
{

```

Has a parent.

```

\int_set:Nn \gls@level { \use:c { glo@\glsdetoklabel{#1}@level } }
\if_empty:NTF \@gls@loclist
{
  \tl_put_right:Ne \l__glossaries_content_tl
  { \exp_not:N \subglossentry { \int_use:N \gls@level } { #1 } { } }
}
{
  \tl_put_right:Ne \l__glossaries_content_tl
  {
    \exp_not:N \subglossentry { \int_use:N \gls@level } { #1 }
    {
      \exp_not:N \glossaryentrynumbers
      {
        \exp_not:N \glsnoidxloclist { \exp_not:V \@gls@loclist }
      }
    }
  }
}
}
{

```

Doesn't have a parent

```

\tl_clear:N \l__glossaries_entry_letter_group_tl
\tl_clear:N \l__glossaries_entry_letter_title_tl
\if_exist:cT
{ glo@ #1 @dtlsgroup }
{
  \tl_set:Ne \l__glossaries_entry_letter_group_tl
  {
    \exp_not:v { glo@ #1 @dtlsgroup }
  }
  \int_compare:nNnT
  { \tl_count:N \l__glossaries_entry_letter_group_tl } = { 2 }
  {
    \tl_set:Ne \l__glossaries_entry_letter_title_tl
    {
      \exp_after:wN \l__glossaries_lettergroup_title:nn
      \l__glossaries_entry_letter_group_tl
    }
    \tl_set:Ne \l__glossaries_entry_letter_group_tl
    { \exp_after:wN \use_i:nn \l__glossaries_entry_letter_group_tl }
  }
}

```

```

        \tl_set_eq:cN
        { l__glossaries_noidx_#1_grouptitle_tl }
        \l__glossaries_entry_letter_title_tl
    }
}
\tl_if_empty:NT \l__glossaries_entry_letter_group_tl
{

```

Doesn't have letter group information. Get this entry's sort key

```

    \letcs{\@gls@sort}{glo@\glsdetoklabel{#1}@sort}%

```

Get the letter group:

```

    \@gls@fetch@lettergrp
    \tl_set_eq:NN
    \l__glossaries_entry_letter_group_tl
    \@glo@thislettergrp
    \tl_clear:c
    { l__glossaries_noidx_#1_grouptitle_tl }
}
\tl_if_eq:NNF
\l__glossaries_entry_prev_letter_group_tl
\l__glossaries_entry_letter_group_tl
{
    \tl_if_empty:NF \l__glossaries_entry_prev_letter_group_tl
    {
        \tl_put_right:Nn \l__glossaries_content_tl
        {
            \glsgroupskip
        }
    }
    \tl_put_right:Ne \l__glossaries_content_tl
    {
        \exp_not:N \glsgroupheading
        { \exp_not:V \l__glossaries_entry_letter_group_tl }
    }
    \tl_set_eq:NN
    \l__glossaries_entry_prev_letter_group_tl
    \l__glossaries_entry_letter_group_tl
}
\tl_if_empty:NTF \@gls@loclist
{
    \tl_put_right:Nn \l__glossaries_content_tl
    {
        \glossentry { #1 } { }
    }
}
{
    \tl_put_right:Ne \l__glossaries_content_tl
    {
        \exp_not:N \glossentry { #1 }
        {

```

```

        \exp_not:N \glossaryentrynumbers
        {
        \exp_not:N \glsnoidxloclist { \exp_not:V \@gls@loclist }
        }
    }
}
}
}
}
}
\cs_new:Nn \__glossaries_lettergroup_title:nn
{
\exp_not:n { #2 }
}
\ExplSyntaxOff

```

`\gls@old@print@noidx@glossary` Original definition (pending deletion):

```

\newcommand*{\@gls@old@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
}%
{%

```

Add section header if there are actually entries defined in this glossary as the document is likely pending a re-run.

```

\glxtrifemptyglossary{\@glo@type}%
{}%
{\glossarysection[\glossarytoctitle]{\glossarytitle}}%
\@gls@noref@warn{\@glo@type}%
}%
}

```

`\glo@grabfirst` This won't work for UTF-8 and inputenc.

```

\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%

```

Sanitize it:

```
\@onelevel@sanitize\@gls@firsttok
```

Fetch the first letter:

```

\expandafter\glo@grabfirst\@gls@firsttok{}{}\@nil
}%
}

```

`\@glo@grabfirst`

```

\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\uccode'#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@='#1\relax
}%
{%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}

```

`\@gls@noidx@do` Handler for list iteration used by `\@print@noidx@glossary`. The argument is the entry label. This only allows one sublevel.

```
\newcommand{\@gls@noidx@do}[1]{%
```

Get this entry's location list

```
\global\letcs{\@gls@loclist}{glo@glsdetoklabel{#1}@loclist}%
```

Does this entry have a parent?

```
\ifglshasparent{#1}%  
{%
```

Has a parent.

```
\gls@level=\csuse{glo@glsdetoklabel{#1}@level}\relax  
\ifdefvoid{\@gls@loclist}  
{%  
  \subglossentry{\gls@level}{#1}{}%  
}%  
{%  
  \subglossentry{\gls@level}{#1}%  
  {%  
    \glsnoidxprenumberlist{#1}%  
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%  
  }%  
}%  
{%
```

Doesn't have a parent Get this entry's sort key

```
\letcs{\@gls@sort}{glo@glsdetoklabel{#1}@sort}%
```

Get the letter group:

```
\@gls@fetch@lettergrp  
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%  
{}%  
{%
```

Do the group header:

```
\ifdefempty{\@gls@currentlettergroup}{}%  
{%
```

The group skip may start a new scope, so make a global assignment.

```
\global\let\@glo@thislettergrp\@glo@thislettergrp  
\glsgroupskip  
}%  
\glsgroupheading{\@glo@thislettergrp}%  
}%  
  
\global\let\@gls@currentlettergroup\@glo@thislettergrp
```

Do this entry:

```
\ifdefvoid{\@gls@loclist}  
{%  
  \glossentry{#1}{}%  
}%  
{%  
  \glossentry{#1}%
```

```

    {%
      \glsnoidxprenumberlist{#1}%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}

```

```
\glsnoidxloclist{<list cs>}
```

\glsnoidxloclist

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
\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}
                    {<location>}

```

Display a location in the location list.

```

\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}

```

```

\@gls@reference
\@gls@reference{<type>}{<label>}{<loc>}

```

Identifies that a reference has been used (for use in the aux file). All entries must be defined in the preamble.

```

\ExplSyntaxOn
\newcommand*\@gls@reference}[3]{%

```

Add to label list

```

  \glsdoifexistsorwarn{#2}%
  {%
    \__glossaries_add_glsref:nnn { #1 } { #2 } { #3 }
  }%
}
\cs_new:Nn \__glossaries_add_glsref:nnn
{
  \__glossaries_add_glsref:nn { #1 } { #2 }

```

Add to location list

```

  \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
  {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
  {}%
  \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
}

```

This may be updated in future to use a sequence variable instead.

```

\cs_new:Nn \__glossaries_add_glsref:nn
{
  \ifcsundef { @glsref@ #1 }
  {
    \csgdef { @glsref@ #1 } { }
    \int_new:c { g__glossaries_gls_ref_ #1 _int }
  }
  { }
\ifinlistcs { #2 } { @glsref@ #1 }
{ }
{
  \listcsgadd { @glsref@ #1 } { #2 }
  \int_gincr:c { g__glossaries_gls_ref_ #1 _int }
}

```

```

    }
  }
\cs_new:Nn \__glossaries_glsref_count:n
{
  \int_if_exist:cTF { g__glossaries_gls_ref_ #1 _int }
  {
    \int_use:c { g__glossaries_gls_ref_ #1 _int }
  }
  { 0 }
}
\ExplSyntaxOff

```

The keys that can be used in the optional argument to `\printglossary` or `\printnoidxglossary` are as follows: The `type` key sets the glossary type.

```
\define@key{printgloss}{type}{\def\@glo@type{#1}}
```

The `title` key sets the title used in the glossary section header. This overrides the title used in `\newglossary`.

```
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotoc@title\relax
}
```

The `toctitle` sets the text used for the relevant entry in the table of contents.

```
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotoc@title\relax
}
```

The `style` key sets the glossary style (but only for the given glossary).

```
\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
  \PackageError{glossaries}%
  {Glossary style ‘#1’ undefined}{}%
}%
{%
  \def\@glossarystyle{\setglossarystyle{#1}}%
}%
}
```

The `numberedsection` key determines if this glossary should be in a numbered section.

```
\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
  \ifcase\gls@numberedsection@nr\relax
  \renewcommand*\@@glossarysecstar{*}%
  \renewcommand*\@@glossaryseclabel{}%
  \or

```

```

\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}

```

The `nogroupskip` key determines whether or not there should be a vertical gap between glossary groups.

```

\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}

```

The `nopostdot` key has the same effect as the package option of the same name.

```

\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}

```

`\GlsEntryCounterLabelPrefix` Make it easier to redefine the label prefix.

```

\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}

```

The conditionals have been moved inside the appropriate commands to make it easier for the user to redefine them in the preamble and selectively switch the counter display on and off. Previously the helper commands were redefined by the `entrycounter` option, which would counteract any earlier customisation.

The `entrycounter` key is the same as the package option but localised to the current glossary.

```

\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{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}}%
    \fi
}
```

```

        \else
        \gls{#1}%
        \fi
    \fi
}

```

`\glsentrycounterlabel` Defines how to display the glossaryentry counter.

```

\newcommand*{\glsentrycounterlabel}{%
\ifglsentrycounter
\theglossaryentry.\space
\fi
}

```

`\glssubentrycounterlabel` Defines how to display the glossarysubentry counter.

```

\newcommand*{\glssubentrycounterlabel}{%
\ifglssubentrycounter
\theglossarysubentry)\space
\fi
}

```

`\glsentryitem` Step and display glossaryentry counter, if appropriate.

```

\newcommand*{\glsentryitem}[1]{%
\ifglsentrycounter
\glsstepentry{#1}\glsentrycounterlabel
\else
\glsresetsubentrycounter
\fi
}

```

`\glssubentryitem` Step and display glossarysubentry counter, if appropriate.

```

\newcommand*{\glssubentryitem}[1]{%
\ifglssubentrycounter
\glsstepsubentry{#1}\glssubentrycounterlabel
\fi
}

```

`theglossary` (*env.*) If the `theglossary` environment has already been defined, a warning will be issued. This environment should be redefined by glossary styles.

```

\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{}{}%
}%
{%
\@gls@warnontheglossdefined
\renewenvironment{theglossary}{}{}%
}

```

The glossary header is given by `\glossaryheader`. This forms part of the glossary style, and must indicate what should appear immediately after the start

of the `theglossary` environment. (For example, if the glossary uses a tabular-like environment, it may be used to set the header row.) Note that if you don't want a header row, the glossary style must redefine `\glossaryheader` to do nothing.

`\glossaryheader`

```
\newcommand*\glossaryheader{}
```

```
\glstarget{\label}{\name}
```

`\glstarget`

Provide user interface to `\@glstarget` to make it easier to modify the glossary style in the document.

```
\newcommand*\glstarget[2]{\@glstarget{\glo@linkprefix#1}{#2}}
```

As from version 3.08, glossary information is now written to the external files using `\glossentry` and `\subglossentry` instead of `\glossaryentryfield` and `\glossarysubentryfield`. The default definition provides backward compatibility for glossary styles that use the old forms.

```
\compatibleglossentry{\label}{\page-list}
```

`\compatibleglossentry`

Removed in v4.50.

```
\glossentry{\label}{\page-list}
```

`\glossentry`

```
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
```

`\glossentryname`

```
\newcommand*\glossentryname[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@glstetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
```

`\Glossentryname`

```
\newcommand*\Glossentryname[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
```

```
\glossentrydesc
  \newcommand*\glossentrydesc[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrydesc{#1}%
    }%
  }
```

```
\Glossentrydesc
  \newcommand*\Glossentrydesc[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
```

```
\glossentrysymbol
  \newcommand*\glossentrysymbol[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
```

```
\Glossentrysymbol
  \newcommand*\Glossentrysymbol[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
```

\compatiblesubglossentry{<level>}{<label>}{<page-list>}

\compatiblesubglossentry

Removed in v4.50

\setglossentrycompatibility

\glossaryentryfield{<label>}{<name>}{<description>}
{<symbol>}{<page-list>}

\glossaryentryfield

This command formerly governed how each entry row should be formatted in the glossary. Now deprecated. Removed in v4.50

`\glossarysubentryfield`

```
\glossarysubentryfield{<level>}{<label>}{<name>}
{<description>}{<symbol>}{<page-list>}
```

This command governs how each subentry should be formatted in the glossary. Glossary styles need to redefine this command. Most of the predefined styles ignore *<symbol>*. The first argument is a number indicating the level. (The level should be greater than or equal to 1.)

Removed in v4.50.

`\subglossentry`

```
\subglossentry{<level>}{<label>}{<page-list>}
```

```
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
```

Within each glossary, the entries form distinct groups which are determined by the first character of the sort key. When using `makeindex`, there will be a maximum of 28 groups: symbols, numbers, and the 26 alphabetical groups A, ..., Z. If you use `xindy` the groups will depend on whatever alphabet is used. This is determined by the language or custom alphabets can be created in the `xindy` style file. The command `\glsgroupskip` specifies what to do between glossary groups. Glossary styles must redefine this command. (Note that `\glsgroupskip` only occurs between groups, not at the start or end of the glossary.)

`\glsgroupskip`

```
\newcommand*{\glsgroupskip}{}
```

Each of the 28 glossary groups described above is preceded by a group heading. This is formatted by the command `\glsgroupheading` which takes one argument which is the *label* assigned to that group (not the title). The corresponding labels are: `glsymbols`, `glsnumbers`, A, ..., Z. Glossary styles must redefine this command. (In between groups, `\glsgroupheading` comes immediately after `\glsgroupskip`.)

`\glsgroupheading`

```
\newcommand*{\glsgroupheading}[1]{} 
```

It is possible to “trick” `makeindex` into treating entries as though they belong to the same group, even if the terms don’t start with the same letter, by modifying the sort key. For example, all entries belonging to one group could be defined so that the sort key starts with an **a**, while entries belonging to another group could be defined so that the sort key starts with a **b**, and so on. If you want each group to have a heading, you would then need to modify the translation control sequences `\glsgetgrouptitle` and `\glsgetgrouplabel` so that the label is translated into the required title (and vice-versa).

```
\glsgetgrouptitle{label}
```

This command produces the title for the glossary group whose label is given by $\langle label \rangle$. 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`, \dots , `Z`. If you want to redefine the group titles, you will need to redefine this command. Languages other than English may produce labels that are non-expandable, so we need to check for that otherwise it will create a “missing `\endcsname` inserted” error.

```
\glsgetgrouptitle
```

```
\newrobustcmd*{\glsgetgrouptitle}[1]{%  
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%  
  \@gls@grptitle  
}
```

`\@gls@getgrouptitle` Gets the group title specified by the label (first argument) and stores in the second argument, which must be a control sequence.

```
\ExplSyntaxOn  
\newcommand*{\@gls@getgrouptitle}[2]{%
```

Note that `\dtl@ifsingle` (provided by `datatool`) is being phased out.

```
\def#2{#1}%  
\regex_match:nnT { ^\A(.|glsnumbers|glssymbols)\Z } { #1 }  
{ \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} }  
}
```

`\@gls@noidx@getgrouptitle` Version for the no-indexing app option:

```
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%  
  \ifcsvoid { l__glossaries_noidx_#1_grouptitle_tl }  
  {  
    \DTLifint{#1}%  
    {\edef#2{\char#1\relax}}%  
    {%  
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%  
    }%  
  }  
  {  
    \tl_set_eq:Nc #2  
    { l__glossaries_noidx_#1_grouptitle_tl }  
  }  
}  
\ExplSyntaxOff
```

```
\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 `\glsnumber` etc.) `\glslink` and `\glsadd` encode the `\glossary` argument so that the relevant counter is set prior to the formatting command.

`\setentrycounter` The prefix is an optional argument because this command originally only had one argument. The optional argument was added in v3.0.

```
\newcommand*\setentrycounter}[2] []{%
  \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.

```

\gls@wrglossdisableanchorcmds
\protected@edef\@gls@tmp{\gls@entrycounter\@glo@counterprefix
  \gls@wrglosslocationtarget{\@gls@x}}%

```

If anything hasn't been fully expanded by this point then it will cause a problem, so sanitize to prevent an error, but it's likely the target won't be valid.

```

  \@onelevel@sanitize\@gls@tmp
  \edef\@gls@x{\noexpand\egroup
    \noexpand\hyperlink{\@gls@tmp}%
    {\noexpand\gls@wrglosslocationtextfmt{\the\toks@}}%
  }%
  \@gls@x
\fi
\else

```

```

    \@gls@ReturnAfterFi{%
      \@gls@removespaces#2\@nil
    }%
  \fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}

```

The following commands will switch to the appropriate font, and create a hyperlink, if hyperlinks are supported. If hyperlinks are not supported, they will just display their argument in the appropriate font.

```

\hyperm
  \newcommand*\hyperm[1]{\textrm{\glsnumber{#1}}}

\hypersf
  \newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}

\hypertt
  \newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}

\hyperbf
  \newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}

\hypermd
  \newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}

\hyperit
  \newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}

\hypersl
  \newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}

\hyperup
  \newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}

\hypersc
  \newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}

\hyperemph
  \newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}

```

2.17 Acronyms

```
\oldacronym[label]{abbrv}{long}{key-val list}
```

\oldacronym

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

$\langle label \rangle$ must only contain alphabetical characters). If $\langle label \rangle$ is omitted, $\langle abbrev \rangle$ is used. This only emulates the syntax of the old `glossary` package. The way the acronyms appear in the list of acronyms is determined by the definition of `\newacronym` and the glossary style.

Note that $\langle label \rangle$ can't have an optional argument if the `xspace` package is loaded. If `xspace` hasn't been loaded then you can do $\langle label \rangle[\langle insert \rangle]$ but you can't do $\langle label \rangle[\langle key-val list \rangle]$. 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[\langle key-val list \rangle]{\langle label \rangle}{\langle abbrev \rangle}{\langle long \rangle}
```

This is a quick way of defining acronyms, using `\newglossaryentry` with the appropriate values. It sets the glossary type to `\acronymtype` which will be `acronym` if the package option `acronym` has been used, otherwise it will be the default glossary. Since `\newacronym` merely calls `\newglossaryentry`, the acronym is treated like any other glossary entry.

If you prefer a different format, you can redefine `\newacronym` as required. The optional argument can be used to override any of the settings.

This is just a stub. It's redefined by `\SetGenericNewAcronym` and by `glossaries-extra`. It used to also be redefined by the obsolete acronym styles that have now been removed. If it is used then it means that the acronym style hasn't been initialised.

```
\newacronym
\newcommand{\newacronym}[4][ ]{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
```

`\gls@newacronym`

```
\newcommand{\gls@newacronym}[4]{%
```

Setting the acronym style will redefine `\newacronym` because it calls `\SetGenericNewAcronym` but if something weird has happened to alter this, redefine `\gls@newacronym` now to prevent an infinite loop.

```
\renewcommand{\gls@newacronym}[4]{%
  \PackageError{glossaries}{incompatible acronym style:
    rollback may be required}%
  {something has gone wrong with \string\newacronym. Try
    \string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}
```

Set up some convenient short cuts. These need to be changed if `\newacronym` is changed (or if the description key is changed).

`\acrpluralsuffix` Plural suffix used by `\newacronym`. This just defaults to `\glspluralsuffix` but is changed to include `\textup` if the smallcaps option is used, so that the suffix doesn't appear in small caps as it doesn't look right. For example, ABCS looks as though the "s" is part of the acronym, but ABCs looks as though the "s" is a plural suffix. Since the entire text `abcs` is set in `\textsc`, `\textup` is need to cancel it out.

```
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

`\glstextup`

```
\newrobustcmd*\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
```

The following are defined for compatibility with version 2.07 and earlier.

`\glsshortkey`

```
\newcommand*\glsshortkey}{short}
```

`\glsshortpluralkey`

```
\newcommand*\glsshortpluralkey}{shortplural}
```

`\glslongkey`

```
\newcommand*\glslongkey}{long}
```

`\glslongpluralkey`

```
\newcommand*\glslongpluralkey}{longplural}
```

Using `\acrlinkfullformat` and `\acrfullformat` is now deprecated. However, these commands are still used in the initial definition of `\acrfull` and `\glsentryfull`. The acronym style now redefines `\glsentryfull`, and

`\acrfull` now uses `\acrfullfmt`, which is redefined by the acronym style. Similar for the case and plural variants. This means that `\acrlinkfullformat` and `\acrfullformat` should now longer be used with v4.50, which will ensure that `\setacronymstyle` is used on the first instance of `\newacronym` if a style hasn't been set.

Outlier cases may occur where someone has defined an entry explicitly with `\newglossaryentry` and set the long and short keys and is using `\glentryfull` or `\acrfull` without a style set. So, for now, the deprecated `\acrlinkfullformat` and `\acrfullformat` are retained even though the remaining obsolete acronym code has been removed. However, they may be removed in a future version.

`\acrlinkfullformat` Format for full links like `\acrfull`. Syntax: `\acrlinkfullformat{<long cs>}{<short cs>}{<options>}{<label>}{<insert>}` Deprecated.

```
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
```

`\acrfullformat` Default full form is *<long>* (*<short>*). Deprecated.

```
\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
```

`\glsspace` Robust space to ensure it's written to the `.glsdefs` file.

```
\newrobustcmd{\glsspace}{\space}
```

The following `\acrfullfmt` set of commands are used by the corresponding `\acrfull` command, and are redefined by the acronym style to match the style's full form.

`\acrfullfmt` No case change full format. This default will be overridden by the acronym style, so this definition shouldn't actually be used any more, but it still needs to be defined so the it doesn't cause a problem when acronym styles redefine it. This is initialised to use the deprecated `\acrlinkfullformat` but will be redefined by the acronym style when it's set.

```
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
```

`\Acrfullfmt` Sentence case full format.

```
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
```

`\ACRfullfmt` All upper case full format.

```
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
```

```

\acrfullplfmt No case change plural full format.
  \newcommand*\acrfullplfmt[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\Acrfullplfmt Sentence case plural full format.
  \newcommand*\Acrfullplfmt[3]{%
    \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
  }

\ACRfullplfmt All upper case plural full format.
  \newcommand*\ACRfullplfmt[3]{%
    \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
  }

\acrfull Full form of the acronym.
  \newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}
  \newcommand*\ns@acrfull[2][ ]{%
    \new@ifnextchar[{\@acrfull{#1}{#2}}%
      {\@acrfull{#1}{#2}[]}%
  }

\@acrfull Low-level macro:
  \def\@acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
    \acrfullfmt{#1}{#2}{#3}%
  }

  Default format for full acronym

\Acrfull
  \newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}
  Add sentence case mapping.
  \glsmfuaddmap{\acrfull}{\Acrfull}
  \newcommand*\ns@Acrfull[2][ ]{%
    \new@ifnextchar[{\@Acrfull{#1}{#2}}%
      {\@Acrfull{#1}{#2}[]}%
  }

  Low-level macro:
  \def\@Acrfull#1#2[#3]{%
    Make it easier for acronym styles to change this:
    \Acrfullfmt{#1}{#2}{#3}%
  }

\ACRfull
  \newrobustcmd*\ACRfull{\@gls@hyp@opt\ns@ACRfull}

```

```

\newcommand*\ns@ACRfull[2] []{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@ACRfull#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```

\ACRfullfmt{#1}{#2}{#3}%
}

```

Plural:

`\acrfullpl`

```

\newrobustcmd*\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}

\newcommand*\ns@acrfullpl[2] []{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```

\acrfullplfmt{#1}{#2}{#3}%
}

```

`\Acrfullpl`

```
\newrobustcmd*\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
```

Add sentence case mapping.

```

\glsmfuaddmap{\acrfullpl}{\Acrfullpl}

\newcommand*\ns@Acrfullpl[2] []{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@Acrfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```

\Acrfullplfmt{#1}{#2}{#3}%
}

```

`\ACRfullpl`

```

\newrobustcmd*\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}

\newcommand*\ns@ACRfullpl[2] []{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2} []}%
}

```

Low-level macro:

```
\def\@ACRfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\ACRfullplfmt{#1}{#2}{#3}%  
}
```

2.18 Predefined acronym styles

`\acronymfont` Used by the acronym styles to apply a font change to the short form.

```
\newcommand{\acronymfont}[1]{#1}
```

`\firstacronymfont` Used by the acronym styles to apply a font change to the short form on first use.

```
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
```

`\acrnameformat` The styles that allow an additional description use `\acrnameformat{<short>}{<long>}` to determine what information is displayed in the name.

```
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
```

Define some tokens used by `\newacronym`:

`\glskeylisttok`

```
\newtoks\glskeylisttok
```

`\glslabeltok`

```
\newtoks\glslabeltok
```

`\glsshorttok`

```
\newtoks\glsshorttok
```

`\glslongtok`

```
\newtoks\glslongtok
```

`\newacronymhook` Provide a hook for `\newacronym`:

```
\newcommand*{\newacronymhook}{}
```

`\SetGenericNewAcronym` New improved version of setting the acronym style.

```
\newcommand*{\SetGenericNewAcronym}{%
```

Change the behaviour of `\Glsentryname` to workaround expansion issues that cause a problem for `\makefirstuc`

```
\let\@Gls@entryname\@Gls@acentryname
```

Change the way acronyms are defined:

```
\renewcommand{\newacronym}[4][ ]{%  
  \ifdefempty{\@glsacronymlists}%  
  {%  
    \def\@glo@type{\acronymtype}%
```

```

\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
}%
{}%
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\protected@edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\expandonce{\acronymentry{##2}}},%
sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
text={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%

longplural={\the\glslongtok\noexpand\glspluralsuffix},%
\GenericAcronymFields,%
\the\glskeylisttok
}%
}%
\@do@newglossaryentry
}%

```

Make sure that `\acrfull` etc reflects the new style:

```

\renewcommand*\acrfullfmt}[3]{%
\glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsuppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsuppercase{\genplacrfullformat{##2}{##3}}}%

```

Make sure that `\glsentryfull` etc reflects the new style:

```

\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}

```

`\GenericAcronymFields` Fields used by `\SetGenericNewAcronym` that can be changed by the acronym style.

```
\newcommand*\GenericAcronymFields{description={\the\glslongtok}}
```

`\acronymentry`

```
\acronymentry{<label>}
```

Display style for the name field in the list of acronyms.

```
\newcommand*\acronymentry[1]{\acronymfont{\glsentryshort{#1}}}
```

`\acronymsort`

```
\acronymsort{<short>}{<long>}
```

Default sort format for acronyms.

```
\newcommand*\acronymsort[2]{#1}
```

`\setacronymstyle`

```
\setacronymstyle{<style name>}
```

```
\newcommand*\setacronymstyle[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{\acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
  }%
}
```

`\newacronymstyle`

```
\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` (*long*) (*short*) acronym style.

```

\newacronymstyle{long-short}%
{%

```

Check for long form in case this is a mixed glossary.

```

  \ifglshaslong{\glslabel}{\glsacfont}{\glsacentryfont}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space

```

```

(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

`long-sp-short` Similar to the previous style but allows the space between the long and short form to be customized.

```

\newacronymstyle{long-sp-short}%
{%
Check for long form in case this is a mixed glossary.
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

`\glsacspace` Space between long and short form for the above style. This uses a non-breakable space if the short form is less than 3em, otherwise it uses a regular space. It's unlikely that the short form will contain any instances of `\gls` so this just uses `\settowidth` instead of `\glsmeasurewidth`.

```

\newcommand*\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}

```

short-long $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\newacronymstyle{short-long}%  
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%  
}%  
{%  
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%  
  \renewcommand*{\genacrfullformat}[2]{%  
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space  
    (\glsentrylong{##1})%  
  }%  
  \renewcommand*{\Genacrfullformat}[2]{%  
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space  
    (\glsentrylong{##1})%  
  }%  
  \renewcommand*{\genplacrfullformat}[2]{%  
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space  
    (\glsentrylongpl{##1})%  
  }%  
  \renewcommand*{\Genplacrfullformat}[2]{%  
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space  
    (\glsentrylongpl{##1})%  
  }%  
  
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%  
  \renewcommand*{\acronymsort}[2]{##1}%  
  \renewcommand*{\acronymfont}[1]{##1}%  
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%  
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%  
}
```

long-sc-short $\langle long \rangle$ ($\textsc{\langle short \rangle}$) acronym style.

```
\newacronymstyle{long-sc-short}%  
{%  
  \GlsUseAcrEntryDispStyle{long-short}%  
}%  
{%  
  \GlsUseAcrStyleDefs{long-short}%  
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%  
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%  
}
```

long-sm-short $\langle long \rangle$ ($\textsmaller{\langle short \rangle}$) acronym style.

```
\newacronymstyle{long-sm-short}%  
{%  
  \GlsUseAcrEntryDispStyle{long-short}%  
}%  
{%
```

```

\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

`sc-short-long` $\langle short \rangle$ ($\backslash\textsc{\langle long \rangle}$) acronym style.

```

\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}

```

`sm-short-long` $\langle short \rangle$ ($\backslash\textsmaller{\langle long \rangle}$) acronym style.

```

\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

`long-short-desc` $\langle long \rangle$ ($\{ \langle short \rangle \}$) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}

```

`long-sp-short-desc` $\langle long \rangle$ ($\{ \langle short \rangle \}$) acronym style that has an accompanying description (which the user needs to supply). The space between the long and short form is given by `\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 <long> (\textsc{<short>}) acronym style that has an accompanying description
(which the user needs to supply).
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (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 <short> ({<long>}) 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 *⟨long⟩* (`\textsc{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

sm-short-long-desc *⟨long⟩* (`\textsmaller{⟨short⟩}`) acronym style that has an accompanying description (which the user needs to supply).

```
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
```

dua *⟨long⟩* only acronym style.

```
\newacronymstyle{dua}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifdefempty\glscustomtext
{%
  \ifglshaslong{\glslabel}%
  {%
    \glsifplural
    {%
```

Plural form:

```
\glscapscase
{%
```

Plural form, don't adjust case:

```
\glsentrylongpl{\glslabel}\glsinsert
}%
{%
```

Plural form, make first letter upper case:

```
\Glsentrylongpl{\glslabel}\glsinsert
```

```

    }%
    {%
Plural form, all caps:
    \glsuppercase
    {\glsentrylongpl{\glslabel}\glsinsert}%
    }%
    }%
    {%
Singular form
    \glsupercase
    {%
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}{\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.
  \ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
}

```

Need to ensure hyperlinks are switched off on first use:

```

\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%

```

Don't use footnotes for \acrfull:

```

\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%

```

Similarly for \glsentryfull etc:

```

\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%

```

```

\renewcommand*\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```

\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glacrpluralsuffix}%
}%

```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymssort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})%
  }
}

```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
}

```

footnote-sm-desc \textsmaller{*short*}\footnote{*long*} acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
}

```

\DefineAcronymSynonyms

```

\newcommand*\DefineAcronymSynonyms}{%

```

Short form

```

\acs
  \let\acs\acrshort

```

First letter uppercase short form

```

\Acs
  \let\Acs\Acrshort

```

Plural short form

```

\acsp
  \let\acsp\acrshortpl

```

First letter uppercase plural short form

```

\Acsp
  \let\Acsp\Acrshortpl

```

Long form

```

\acl
  \let\acl\aclong

```

Plural long form

`\aclp`
`\let\aclp\acrlongpl`

First letter upper case long form

`\Acl`
`\let\Acl\Acrlong`

First letter upper case plural long form

`\Aclp`
`\let\Aclp\Acrlongpl`

Full form

`\acf`
`\let\acf\acrfull`

Plural full form

`\acfp`
`\let\acfp\acrfullpl`

First letter upper case full form

`\Acf`
`\let\Acf\Acrfull`

First letter upper case plural full form

`\Acfp`
`\let\Acfp\Acrfullpl`

Standard form

`\ac`
`\let\ac\gls`

First upper case standard form

`\Ac`
`\let\Ac\Gls`

Standard plural form

`\acp`
`\let\acp\glspl`

Standard first letter upper case plural form

`\Acp`
`\let\Acp\Glspl`

}

Define synonyms if required

```
\ifglsacrshortcuts
\DefineAcronymSynonyms
\fi
```

`\SetDefaultAcronymDisplayStyle` Sets the default acronym display style for given glossary. Used by `glossaries-extra` when restoring `\newacronym`.

```
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{\glsentryfmt}%
}
```

The following commands for setting the acronym style are now deprecated.

`\DefaultNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\glslongtok` and `\glskeylisttok`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\noexpand@assign@default@plurals` `\@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@gls@@default@acr@plural`

`\gls@@default@acr@firstplural`

`\gls@@default@acr@descplural`

`\SetDefaultAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\acrfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote-#1-#2-#3}
```

`\acrlinkfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink[#1]#2-#3}%
}
```

`\acrnofootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

```
\newcommand*{\acrnofootnote}[3]{%
\footnote{#3}%
}
```

`\FootnoteAcronymDisplayStyle` Sets the acronym display style for given glossary for the description and footnote combination. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\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.

`\DescriptionDUAAcronymStyle` Description, don't use acronym and no footnote. Note that the short form is stored in the `symbol` key, so if the short form needs to be displayed in the glossary, use a style 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 `\acronymformat` 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@glstetoklabel{#1}@parent\endcsname
}
```

`\showglolevel` `\showglolevel{<label>}`

```
\newcommand*{\showglolevel}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@level\endcsname
}
```

`\showglotext` `\showglotext{<label>}`

```
\newcommand*{\showglotext}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
}
```

`\showgloplural` `\showgloplural{<label>}`

```
\newcommand*{\showgloplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
}
```

`\showglofirst` `\showglofirst{<label>}`

```
\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
```

`\showglofirstpl` `\showglofirstpl{<label>}`

```
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
```

\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
}
```

\showglouserii

```
\showglouserii{<label>}
```

```
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
}
```

\showglouseriii

```
\showglouseriii{<label>}
```

```
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
}
```

\showglouseriiii

```
\showglouseriiii{<label>}
```

```
\newcommand*{\showglouseriiii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriiii\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}@useriv\endcsname  
}
```

\showglouservi

```
\showglouservi{<label>}
```

```
\newcommand*{\showglouservi}[1]{%  
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\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@glstetoklabel{#1}@sort\endcsname  
}
```

`\showglosymbol`

```
\showglosymbol{<label>}
```

```
\newcommand*{\showglosymbol}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@symbol\endcsname  
}
```

`\showglosymbolplural`

```
\showglosymbolplural{<label>}
```

```
\newcommand*{\showglosymbolplural}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@symbolplural\endcsname  
}
```

`\showgloshort`

```
\showgloshort{<label>}
```

```
\newcommand*{\showgloshort}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@short\endcsname  
}
```

`\showglolong`

```
\showglolong{<label>}
```

```
\newcommand*{\showglolong}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@long\endcsname  
}
```

`\showgloindex`

```
\showgloindex{<label>}
```

```
\newcommand*{\showgloindex}[1]{%  
  \expandafter\show\csname glo@glstetoklabel{#1}@index\endcsname  
}
```

`\showgloflag`

```
\showgloflag{<label>}
```

```
\newcommand*{\showgloflag}[1]{%  
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname  
}
```

`\showgloloclist`

```
\showgloloclist{<label>}
```

```
\newcommand*{\showgloloclist}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname  
}
```

`\showglofield`

```
\showglofield{<label>}{<field>}
```

```
\newcommand*{\showglofield}[2]{%  
  \csshow{glo@\glsdetoklabel{#1}@#2}%  
}
```

`\showacronymlists`

```
\showacronymlists
```

Show list of glossaries that have been flagged as a list of acronyms.

```
\newcommand*{\showacronymlists}{%  
  \show\@glsacronymlists  
}
```

`\showglossaries`

```
\showglossaries
```

Show list of defined glossaries.

```
\newcommand*{\showglossaries}{%  
  \show\@glo@types  
}
```

`\showglossaryin`

```
\showglossaryin{<glossary-label>}
```

Show the ‘in’ extension for the given glossary.

```
\newcommand*{\showglossaryin}[1]{%  
  \expandafter\show\csname @glo@#1@in\endcsname  
}
```

`\showglossaryout`

```
\showglossaryout{<glossary-label>}
```

Show the ‘out’ extension for the given glossary.

```
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
```

`\showglossarytitle`

```
\showglossarytitle{<glossary-label>}
```

Show the title for the given glossary.

```
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
```

`\showglossarycounter`

```
\showglossarycounter{<glossary-label>}
```

Show the counter for the given glossary.

```
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
```

`\showglossaryentries`

```
\showglossaryentries{<glossary-label>}
```

Show the list of entry labels for the given glossary.

```
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}
```

2.21 Compatibility with version 2.07 and below

The backward-compatibility options have been removed in version 4.50. Use the rollback feature instead if backward-compatibility is required. Bear in mind that v2.07 was released 2010-07-10 and v3.07 was released 2013-07-05. Any old documents from that long ago will probably have problems with recent kernels.

In order to fix some bugs in v3.0, it was necessary to change the way information is written to the `glo` file, which also meant a change in the format of the Xindy style file. The compatibility option is meant for documents that use a customised Xindy style file with `\noist`. With the compatibility option, hopefully `xindy` will still be able to process the old document, but the bugs will remain. The issues in versions 2.07 and below:

- With `xindy`, the counter used by the entry was hard-coded into the Xindy style file. This meant that you couldn’t use the counter to swap counters.

- With both `xindy` and `makeindex`, if used with `hyperref` and `\theH<counter>` was different to `\thecounter`, the link in the location number would be undefined.

3 Glossary Styles

3.1 Glossary hyper-navigation definitions (glossary-hypernav.sty)

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-hypernav-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-hypernav-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossary-hypernav-2022-11-03.sty}
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2025/03/12 v4.57 (NLCT)]
```

The commands defined in this package are provided to help navigate around the groups within a glossary (see [subsection 2.16.](#)) `\printglossary` (and `\printglossaries`) set `\@glo@type` to the label of the current glossary. This is used to create a unique hypertarget in the event of multiple glossaries.

```
\glsnavhyperlink[<type>]{<label>}{<text>}
```

This command makes `<text>` a hyperlink to the glossary group whose label is given by `<label>` for the glossary given by `<type>`.

`\glsnavhyperlink` The optional argument is the glossary type and can be used to make the group label unique in case the same group occurs in multiple glossaries. The optional argument was `\@glo@type` but is now `\currentglossary` because that's less likely to change within the glossary. It's unlikely that the optional argument will be used to set it to anything other than `\currentglossary` but it makes it possible to hyperlink to the group from outside of the glossary.

```
\newcommand*\glsnavhyperlink{3}[\currentglossary]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
```

`\glsnavhyperlinkname` Expands to the hypertarget name. The first argument is the glossary type. The second argument is the group label.

```
\newcommand*\glsnavhyperlinkname{2}{glsn:#1@#2}
```

Version 4.53: reimplemented list of hypertarget groups as a sequence. Previously, the targets were stored as comma-separated lists in `\@gls@hypergroupplist@<glossary>` (where `<glossary>` is the glossary label). They are now stored in a sequence `\g_glossaries_<glossary>_hypergroups_seq`.

```
\ExplSyntaxOn
```

```
\glossaries_add_hypergroup:nn{<glossary>}{<target>}
```

Add a target to the sequence unless it's already contained in it:

```
\cs_new:Npn \glossaries_add_hypergroup:nn #1 #2
{
  \seq_if_exist:cTF
  { g_glossaries_ #1 _hypergroups_seq }
  {
    \seq_if_in:cnF
    { g_glossaries_ #1 _hypergroups_seq } { #2 }
    {
      \seq_gput_right:cn
      { g_glossaries_ #1 _hypergroups_seq }
      { #2 }
    }
  }
  {
    \seq_new:c { g_glossaries_ #1 _hypergroups_seq }
    \seq_gput_right:cn
    { g_glossaries_ #1 _hypergroups_seq }
    { #2 }
  }
}
\cs_generate_variant:Nn \glossaries_add_hypergroup:nn { nx }
```

Check if the given target has been added to the sequence to avoid duplicates.

```
\cs_new:Npn \glossaries_if_hypergroup:nnTF #1 #2 #3 #4
{
  \seq_if_exist:cTF
  { g_glossaries_ #1 _hypergroups_seq }
  {
    \seq_if_in:cnTF
    { g_glossaries_ #1 _hypergroups_seq } { #2 }
    { #3 } { #4 }
  }
  { #4 }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnTF { nxTF }
```

Only true part:

```
\cs_new:Npn \glossaries_if_hypergroup:nnT #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { #3 } { }
}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnT { nxT }
```

Only false part:

```
\cs_new:Npn \glossaries_if_hypergroup:nnF #1 #2 #3
{
  \glossaries_if_hypergroup:nnTF { #1 } { #2 } { } { #3 }
```

```

}
\cs_generate_variant:Nn \glossaries_if_hypergroup:nnF { nxF }

```

```
\glsnavhypertarget[⟨type⟩]{⟨label⟩}{⟨text⟩}
```

`\glsnavhypertarget`

This command makes *⟨text⟩* a hypertarget for the glossary group whose label is given by *⟨label⟩* in the glossary given by *⟨type⟩*. If *⟨type⟩* is omitted, `\gls@type` is used which is set by `\printglossary` to the current glossary label. The optional argument complicates things for `glossaries-extra`.

```
\newcommand*\glsnavhypertarget{\protect\@gls@navhypertarget}
```

`\@gls@navhypertarget`

```

\newcommand*\@gls@navhypertarget}[3][\currentglossary]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}

```

`\glsnavhypergroupdotarget` Provided to make it easier for `glossaries-extra` to adjust the target.

```

\newcommand{\glsnavhypergroupdotarget}[3]{
  \glsstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
}

```

The actual code is now in an internal command that doesn't have an optional argument, which makes it easier to save and restore the original behaviour.

`\@glsnavhypertarget`

```
\newcommand*\@glsnavhypertarget}[3]{%
```

Add this group to the aux file for re-run check.

```
\protected@write\@auxout{\string\@gls@hypergroup{#1}{#2}}%
```

Add the target.

```
\glsnavhypergroupdotarget { #1 } { #2 } { #3 }
```

Check list of known groups to determine if a re-run is required.

```

\glossaries_if_hypergroup:nxF { #1 } { #2 }
{

```

This group was not included in the list, so issue a warning.

```

\GlossariesWarningNoLine{Navigation ~ panel ~
  for ~ glossary ~ type ~ '#1' ^^Jmissing ~ group ~ '#2'}%

```

```
\gdef\gls@hypergrouprerun{%
```

```

\GlossariesWarningNoLine{Navigation ~ panel ~
  has ~ changed. ~ Rerun ~ LaTeX}%

```

```

}
}

```

`\gls@hypergrouprerun` Give a warning at the end if re-run required

```

\let\gls@hypergrouprerun\relax
\AtEndDocument{\gls@hypergrouprerun}

```

`\@gls@hypergroup` This adds to (or creates) the command `\@gls@hypergrouplist@{glossary type}` which lists all groups for a given glossary, so that the navigation bar only contains those groups that are present. However it requires at least 2 runs to ensure the information is up-to-date.

```
\newcommand*{\@gls@hypergroup}[2]{
  \glossaries_add_hypergroup:nx { #1 } { #2 }
}
```

The `\glsnavigation` command displays a simple glossary group navigation. The symbol and number elements are defined separately, so that they can be suppressed if need be. (In earlier versions this command will produce a link to all 28 groups, but some groups may not be defined if there are groups that do not contain any terms, in which case you will get an undefined hyperlink warning. Version 1.14 changed this to only use labels for groups that are present.) Now for the whole navigation bit:

```
\glsnavigation
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \seq_map_inline:cn { g_glossaries_ \currentglossary _hypergroups_seq }
  {
    \@gls@between
    \glsnavigationitem { ##1 }
    \let\@gls@between\glshypernavsep
  }%
}
```

```
\glsnavigationitem
\newcommand{\glsnavigationitem}[1]{%
  \@gls@getgrouptitle { #1 } { \@gls@grptitle }
  \exp_args:Nno \glsnavhyperlink { #1 } { \@gls@grptitle }
}
```

`\glshypernavsep` Separator for the hyper navigation bar.

```
\newcommand*{\glshypernavsep}{\space\textbar\space}
```

The `\glssymbolnav` produces a simple navigation set of links for just the symbol and number groups. This used to be used at the start of `\glsnavigation`. This command is no longer needed.

```
\glssymbolnav
\newcommand*{\glssymbolnav}{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}
\ExplSyntaxOff
```

3.2 In-line Style (glossary-inline.sty)

This defines an in-line style where the entries are comma-separated with just the name and description displayed.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-inline-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossary-inline-2021-11-01.sty}
```

```
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-inline}[2025/03/12 v4.57 (NLCT)]
```

`inline` Define the inline style.

```
\newglossarystyle{inline}{%
```

Start of glossary sets up first empty separator between entries. (This is then changed by `\glossentry`)

```
\renewenvironment{theglossary}%  
  {%  
    \def\gls@inlinesep{}%  
    \def\gls@inlinesubsep{}%  
    \def\gls@inlinepostchild{}%  
  }%  
  {\glspostinline}%
```

No header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings (if heading is required, add `\glsinlinedopostchild` to start definition in case heading follows a child entry):

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Just display separator followed by name and description:

```
\renewcommand{\glossentry}[2]{%  
  \glsinlinedopostchild  
  \gls@inlinesep  
  \glsentryitem{##1}%  
  \glsinlinenameformat{##1}{%  
    \glossentryname{##1}%  
  }%  
  \ifglsdescsuppressed{##1}%  
  {%  
    \glsinlineemptydescformat  
    {%  
      \glossentrysymbol{##1}%  
    }%  
    {%  
      ##2%  
    }%  
  }%  
}%
```

```

{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\glsinlineifhaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%

```

Sub-entries display description:

```

\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%

```

Nothing special between groups:

```

\renewcommand*{\glsgroupskip}{}%
}

```

`\glsinlineifhaschildren`

```

\newcommand{\glsinlineifhaschildren}[3]{%
  \ifglshaschildren{#1}{#2}{#3}%
}

```

`\glsinlinedopostchild`

```

\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}

```

`\glsinlineseparator` Separator to use between entries.

```

\newcommand*{\glsinlineseparator}{;\space}

```

`\glsinlinesubseparator` Separator to use between sub-entries.

```

\newcommand*{\glsinlinesubseparator}{,\space}

```

`\glsinlineparentchildseparator` Separator to use between parent and children.

```

\newcommand*{\glsinlineparentchildseparator}{:\space}

```

`\glsinlinepostchild` Hook to use between child and next entry

```
\newcommand*\glsinlinepostchild{}
```

`\glspostinline` Terminator for inline glossary.

```
\newcommand*\glspostinline{\glspostdescription\space}
```

`\glsinlinenameformat` Formats the name of the entry (first argument label, second argument name):

```
\newcommand*\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
```

`\glsinlinedescformat` Formats the entry’s description, symbol and location list:

```
\newcommand*\glsinlinedescformat}[3]{\space#1}
```

`\glsinlineemptydescformat` Formats the entry’s symbol and location list when the description is empty:

```
\newcommand*\glsinlineemptydescformat}[2]{}
```

`\glsinlinesubnameformat` Formats the name of the subentry (first argument label, second argument name):

```
\newcommand*\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
```

`\glsinlinesubdescformat` Formats the subentry’s description, symbol and location list:

```
\newcommand*\glsinlinesubdescformat}[3]{#1}
```

3.3 List Style (`glossary-list.sty`)

The `glossary-list` style file defines glossary styles that use the `description` environment. Note that since the entry name is placed in the optional argument to the `\item` command, it will appear in a bold font by default.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-list-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossary-list-2021-11-01.sty}
```

```
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-list}[2025/03/12 v4.57 (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[\glstryitem{##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
  \glsentryitem{##2}%
  \glstarget{##2}{\strut\glossentrydesc{##2}\glspostdescription\space ##3}%
}
```

altlistgroup The `altlistgroup` glossary style is like the `altlist` style, but the glossary groups have headings.

```
\newglossarystyle{altlistgroup}{%
```

Base it on the `altlist` style:

```
\setglossarystyle{altlist}%
```

Each group has a heading:

```
\renewcommand*\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}%
```

altlisthypergroup The `altlisthypergroup` glossary style is like the `altlistgroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{altlisthypergroup}{%
```

Base it on the `altlist` style:

```
\setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
\renewcommand*\glossaryheader}{%
  \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a `hypertarget`:

```
\renewcommand*\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}%
```

listdotted The `listdotted` glossary style was supplied by Axel Menzel. I've modified it slightly so that the distance from the start of the name to the end of the dotted line is specified by `\glslistdottedwidth`. Note that this style ignores the page numbers as well as the symbol. Sub-entries are displayed in the same way as top-level entries.

```
\newglossarystyle{listdotted}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Each main (level 0) entry starts a new item:

```
\renewcommand*{\glossentry}[2]{%
  \item[\makebox[\glslistdottedwidth][l]{%
    \glsentryitem{##1}%
    \glstarget{##1}{\glossentryname{##1}}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
}
```

Sub entries have the same format as main entries:

```
\renewcommand*{\subglossentry}[3]{%
  \item[\makebox[\glslistdottedwidth][l]{%
    \glsesubentryitem{##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{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2025/03/12 v4.57 (NLCT)]
```

Requires the `longtable` package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. (There's a chance that the user may specify `nolong` and then load `glossary-long` later, in which case `\glsdescwidth` may have already been defined by `glossary-super`. The same goes for `\glspagelistwidth`.)

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`long` The long glossary style command which uses the `longtable` environment:

```
\newglossarystyle{long}{%
```

Use `longtable` with two columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{glsdescwidth}]}%
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

longborder The longborder style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longborder}{%
```

Base it on the glostylelong style:

```
\setglossarystyle{long}{%
```

Use longtable with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%  
  \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%  
}
```

longheader The longheader style is like the long style but with a header:

```
\newglossarystyle{longheader}{%
```

Base it on the glostylelong style:

```
\setglossarystyle{long}{%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%  
  \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%  
}
```

longheaderborder The longheaderborder style is like the long style but with a header and border:

```
\newglossarystyle{longheaderborder}{%
```

Base it on the glostylelongborder style:

```
\setglossarystyle{longborder}{%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%  
  \hline\bfseries \entryname & \bfseries  
  \descriptionname\tabularnewline\hline  
  \endhead  
  \hline\endfoot}%  
}
```

long3col The long3col style is like long but with 3 columns

```
\newglossarystyle{long3col}{%
```

Use a longtable with 3 columns:

```
\renewenvironment{theglossary}{%  
  {\begin{longtable}{lp{\glsdescwidth}p{\glspagerlistwidth}}}%  
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgruppeheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsgroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi
}
```

`long3colborder` The `long3colborder` style is like the `long3col` style but with a border:

```
\newglossarystyle{long3colborder}{%
```

Base it on the `glostylelong3col` style:

```
\setglossarystyle{long3col}%
```

Use a `longtable` with 3 columns with vertical lines around them:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
```

`long3colheader` The `long3colheader` style is like `long3col` but with a header row:

```
\newglossarystyle{long3colheader}{%
```

Base it on the `glostylelong3col` style:

```
\setglossarystyle{long3col}%
```

Set the table's header:

```
\renewcommand*\glossaryheader{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`long3colheaderborder` The `long3colheaderborder` style is like the above but with a border

```
\newglossarystyle{long3colheaderborder}{%
```

Base it on the `glostylelong3colborder` style:

```
\setglossarystyle{long3colborder}%
```

Set the table's header and add horizontal line at table's foot:

```
\renewcommand*{\glossaryheader}{%  
  \hline  
  \bfseries\entryname&\bfseries\descriptionname&  
  \bfseries\pagelistname\tabularnewline\hline\endhead  
  \hline\endfoot}%
```

```
}
```

`long4col` The `long4col` style has four columns where the third column contains the value of the associated symbol key.

```
\newglossarystyle{long4col}{%
```

Use a `longtable` with 4 columns:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{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]{%  
  \glstarget{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} &  
  \glossentrysymbol{##1} &  
  ##2\tabularnewline  
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  \glossentrysymbol{##2} & ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip  
\renewcommand*{\glsgroupskip}{%}
```

```

\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}

```

`long4colheader` The `long4colheader` style is like `long4col` but with a header row.

```

\newglossarystyle{long4colheader}{%
Base it on the glostylelong4col style:
\setglossarystyle{long4col}%
Table has a header:
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}

```

`long4colborder` The `long4colborder` style is like `long4col` but with a border.

```

\newglossarystyle{long4colborder}{%
Base it on the glostylelong4col style:
\setglossarystyle{long4col}%
Use a longtable with 4 columns surrounded by vertical lines:
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|l|l|l|}}%
  {\end{longtable}}%
Add horizontal lines to the head and foot of the table:
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}

```

`long4colheaderborder` The `long4colheaderborder` style is like the above but with a border.

```

\newglossarystyle{long4colheaderborder}{%
Base it on the glostylelong4col style:
\setglossarystyle{long4col}%
Use a longtable with 4 columns surrounded by vertical lines:
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|l|l|l|}}%
  {\end{longtable}}%
Add table header and horizontal line at the table's foot:
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}

```

`altlong4col` The `altlong4col` style is like the `long4col` style but can have multiline descriptions and page lists.

```
\newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
\newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
\setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
\newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
\setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}%  
  {\end{longtable}}%  
}
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `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}{|lp{\glsdescwidth}|lp{\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{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2025/03/12 v4.57 (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.

```
\ifglsgnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`long4col-booktabs` The `long4col-booktabs` style is similar to the `long4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long4col` style as a base.

```
\setglossarystyle{long4col}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
  \toprule \bfseries \entryname &
  \bfseries \descriptionname &
  \bfseries \symbolname &
  \bfseries \pagelistname
  \tabularnewline\midrule\endhead
  \bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

`altlong4col-booktabs` The `altlong4col-booktabs` style is similar to the `altlong4colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlong4col-booktabs}{%
```

The patch `\glspatchLToutput` is already applied in `long4col-booktabs` and so doesn't need to be here.

```
\glspatchLToutput
```

Use the `long4col-booktabs` style as a base.

```
\setglossarystyle{long4col-booktabs}%
```

Change the column specifications:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
```

Ragged styles.

`longragged-booktabs` The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long-booktabs` style as a base.

```
\setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}}}%
  {\end{longtable}}%
}
```

`longragged3col-booktabs` The `longragged3col-booktabs` style is similar to the `longragged3col` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long3col-booktabs style as a base.

```
\setglossarystyle{long3col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}%  
    >{\raggedright}p{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

`altlongragged4col-booktabs` The `altlongragged4col-booktabs` style is similar to the `altlongragged4col` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlongragged4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `altlong4col-booktabs` style as a base.

```
\setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%  
    >{\raggedright}p{\glspagelistwidth}}%  
  {\end{longtable}}%  
}
```

```
\glsLTpenaltycheck
```

```
\newcommand*{\glsLTpenaltycheck}{%  
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi  
}
```

```
\glspenaltygroupskip
```

```
\newcommand{\glspenaltygroupskip}{%  
  \noalign{\penalty-50\vskip\normalbaselineskip}}
```

`\glsrestoreLToutput` Provide a way of restoring `\LT@output` for the user.

```
\let\@gls@org@LT@output\LT@output  
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
```

This is David's patch, but I've replaced the hard-coded values with `\glsLTpenaltycheck` to make it easier to adjust.

`\glspatchLToutput`

```
\newcommand*\glspatchLToutput}{%
\renewcommand*\LT@output}{%
  \ifnum\outputpenalty <-\@Mi
  \ifnum\outputpenalty > -\LT@end@pen
    \LT@err{floats and marginpars not allowed in a longtable}\@ehc
  \else
    \setbox\z@\vbox{\unvbox\@cclv}%
    \ifdim \ht\LT@lastfoot>\ht\LT@foot
      \dimen@pagegoal
      \advance\dimen@-\ht\LT@lastfoot
      \ifdim\dimen@<\ht\z@
        \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \setbox\z@\vbox{\box\LT@head\glslTpenaltycheck}%
      \fi
    \fi
    \global\@colroom\@colht
    \global\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{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2025/03/12 v4.57 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the longtable package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```
\@ifundefined{glsdescwidth}{%  
  \newlength{glsdescwidth}  
  \setlength{glsdescwidth}{0.6\hsize}  
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%  
  \newlength{glspagelistwidth}  
  \setlength{glspagelistwidth}{0.1\hsize}  
}{}
```

`longragged` The longragged glossary style is like the long but uses ragged right formatting for the description column.

```
\newglossarystyle{longragged}{%
```

Use longtable with two columns:

```
\renewenvironment{theglossary}{%  
  {\begin{longtable}[1>{\raggedright}p{glsdescwidth}}}%  
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupeheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%  
  \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1}\glspostdescription\space ##2%  
  \tabularnewline  
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2}%  
  \glspostdescription\space ##3%  
  \tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{ }%
\else
  \renewcommand*\glsgroupskip{ & \tabularnewline}%
\fi
}
```

`longraggedborder` The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
```

`longraggedheader` The `longraggedheader` style is like the `longragged` style but with a header:

```
\newglossarystyle{longraggedheader}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Set the table's header:

```
\renewcommand*\glossaryheader{%
  \bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\endhead}%
}
```

`longraggedheaderborder` The `longraggedheaderborder` style is like the `longragged` style but with a header and border:

```
\newglossarystyle{longraggedheaderborder}{%
```

Base it on the `glostylelongraggedborder` style:

```
\setglossarystyle{longraggedborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*\glossaryheader{%
  \hline\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`longragged3col` The `longragged3col` style is like `longragged` but with 3 columns

```
\newglossarystyle{longragged3col}{%
  Use a longtable with 3 columns:
  \renewenvironment{theglossary}%
    {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`longragged3colborder` The `longragged3colborder` style is like the `longragged3col` style but with a border:

```
\newglossarystyle{longragged3colborder}{%
  Base it on the glostylelongragged3col style:
  \setglossarystyle{longragged3col}%
  Use a longtable with 3 columns with vertical lines around them:
  \renewenvironment{theglossary}%
    {\begin{longtable}[|l|>{\raggedright}p{\glsdescwidth}||%
      >{\raggedright}p{\glspagelistwidth}||}%
    {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`longragged3colheader` The `longragged3colheader` style is like `longragged3col` but with a header row:

```
\newglossarystyle{longragged3colheader}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`longragged3colheaderborder` The `longragged3colheaderborder` style is like the above but with a border

```
\newglossarystyle{longragged3colheaderborder}{%
```

Base it on the `glostylelongragged3colborder` style:

```
\setglossarystyle{longragged3colborder}%
```

Set the table's header and add horizontal line at table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

`altlongragged4col` The `altlongragged4col` style is like the `altlong4col` style defined in the `glossary-long` package, except that ragged right formatting is used for the description and page list columns.

```
\newglossarystyle{altlongragged4col}{%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[1]>{\raggedright}p{\glsdescwidth}l%
   >{\raggedright}p{\glspagelistwidth}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glosseentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glosseentryname{##1}} &
```

```

    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%

```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```

\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%

```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{ & & \tabularnewline}%
\fi
}

```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```

\newglossarystyle{altlongragged4colheader}{%

```

Base it on the `glostylealtlongragged4col` style:

```

  \setglossarystyle{altlongragged4col}%

```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```

\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%

```

Table has a header:

```

\renewcommand*\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}

```

`altlongragged4colborder` The `altlongragged4colborder` style is like `altlongragged4col` but with a border.

```

\newglossarystyle{altlongragged4colborder}{%

```

Base it on the `glostylealtlongragged4col` style:

```

  \setglossarystyle{altlongragged4col}%

```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```

\renewenvironment{theglossary}%

```

```

\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
>{\raggedright}p{\glspagelistwidth}|}}%
\end{longtable}}%

```

Add horizontal lines to the head and foot of the table:

```

\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}

```

`altlongragged4colheaderborder` The `altlongragged4colheaderborder` style is like the above but with a header as well as a border.

```

\newglossarystyle{altlongragged4colheaderborder}{%

```

Base it on the `glostylealtlongragged4col` style:

```

\setglossarystyle{altlongragged4col}%

```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```

\renewenvironment{theglossary}%
{\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
>{\raggedright}p{\glspagelistwidth}|}}%
{\end{longtable}}%

```

Add table header and horizontal line at the table's foot:

```

\renewcommand*{\glossaryheader}{%
\hline\bfseries\entryname&\bfseries\descriptionname&
\bfseries \symbolname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}

```

3.7 Glossary Styles using `multicol` (`glossary-mcols.sty`)

The `glossary-mcols` style file defines glossary styles that use the `multicol` package. These use the tree-like glossary styles in a `multicol` environment.

```

\NeedsTeXFormat{LaTeX2e}

```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-mcols-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-mcols-2021-11-01.sty}
\DeclareCurrentRelease{v4.57}{2025-03-12}

```

Declare package:

```

\ProvidesPackage{glossary-mcols}[2025/03/12 v4.57 (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}{\glsgrouptitle{##1}}}%
  \indexspace}%
}
```

`mcolindexspannav` Similar to `mcolindexhypergroup`, but puts the navigation line in the optional argument of `multicols`.

```

\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%

    \let\item\glstreeitem}%
  \end{multicols}}%

```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```

\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}

```

mcoltree Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```

\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  \end{multicols}}%
}

```

mcoltreegroup Like the mcoltree style but the glossary groups have headings.

```

\newglossarystyle{mcoltreegroup}{%
  Base it on the glostylemcoltree style:
  \setglossarystyle{mcoltree}%

  Each group has a heading (in bold) followed by a vertical gap):
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

mcoltreehypergroup The mcoltreehypergroup style is like the treegroup style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcoltreehypergroup}{%
  Base it on the glostylemcoltree style:
  \setglossarystyle{mcoltree}%

```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreesspannav` Similar to the `mcoltreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreesspannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreenoname` Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
```

`mcoltreenonamegroup` Like the `mcoltreenoname` style but the glossary groups have headings.

```
\newglossarystyle{mcoltreenonamegroup}{%
Base it on the glostylemcoltreenoname style:
\setglossarystyle{mcoltreenoname}%
}
```

Give each group a heading:

```
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`mcoltreenamehypergroup` The `mcoltreenamehypergroup` style is like the `mcoltreenamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcoltreenamehypergroup}{%
```

Base it on the `glostylemcoltreename` style:

```
\setglossarystyle{mcoltreename}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreenamespannav` Similar to the `mcoltreenamehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreenamespannav}{%
\setglossarystyle{treename}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcolalttree` Multi-column index style. Same as the `alttree`, but puts the glossary in multiple columns.

```
\newglossarystyle{mcolalttree}{%
\setglossarystyle{alttree}%
\renewenvironment{theglossary}%
{%
```

```

        \begin{multicols}{\glsmcols}
        \def\@gls@prevlevel{-1}%
        \mbox{}\par
    }%
    {\par\end{multicols}}%
}

```

`mcolalttreegroup` Like the `mcolalttree` style but the glossary groups have headings.

```

\newglossarystyle{mcolalttreegroup}{%
Base it on the glostylemcolalttree style:
\setglossarystyle{mcolalttree}%
Give each group a heading.
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}

```

`mcolalttreehypergroup` The `mcolalttreehypergroup` style is like the `mcolalttreegroup` style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcolalttreehypergroup}{%
Base it on the glostylemcolalttree style:
\setglossarystyle{mcolalttree}%
Put the navigation links in the header
\renewcommand*\glossaryheader{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
Put a hypertarget at the start of each group
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

`mcolalttreespannav` Similar to the `mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```

\newglossarystyle{mcolalttreespannav}{%
\setglossarystyle{alttree}%
\renewenvironment{theglossary}%

```

```

{%
  \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
  \def\@gls@prevlevel{-1}%
  \mbox{}\par
}%
{\par\end{multicols}}%

```

Put a hypertarget at the start of each group

```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

3.8 Glossary Styles using supertabular environment (glossary-super.sty)

The glossary styles defined in the glossary-super package use the supertabular environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossary-super-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-super-2021-11-01.sty}
\DeclareCurrentRelease{v4.57}{2025-03-12}

```

Declare package:

```
\ProvidesPackage{glossary-super}[2025/03/12 v4.57 (NLCT)]
```

Requires the supertabular package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if `glossary-long` has been loaded.

```

\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{glsdescwidth}{0.6\hsize}
}{}

```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if `glossary-long` has been loaded.

```

\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{glspagelistwidth}{0.1\hsize}
}{}

```

super The **super** glossary style uses the **supertabular** environment (it uses lengths defined in the **glossary-long** package.)

```
\newglossarystyle{super}{%
```

Put the glossary in a **supertabular** environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{ }\tabletail{ }}%
  \begin{supertabular}{lp{\glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for **nogroupskip** must occur outside **\glsgroupskip** (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
```

superborder The **superborder** style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superborder}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
```

```

        {\tablehead{\hline}\tabletail{\hline}%
         \begin{supertabular}{|l|p{\glsdescwidth}|}%
         {\end{supertabular}}%
    }

```

superheader The superheader style is like the super style, but with a header:

```
\newglossarystyle{superheader}{%
```

Base it on the glostylesuper style:

```
\setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and no tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname &
   \bfseries \descriptionname\tabularnewline}%
   \tabletail{}}%
  {\begin{supertabular}{lp{\glsdescwidth}}%
   {\end{supertabular}}%
}

```

superheaderborder The superheaderborder style is like the super style but with a header and border:

```
\newglossarystyle{superheaderborder}{%
```

Base it on the glostylesuper style:

```
\setglossarystyle{super}%
```

Put the glossary in a supertabular environment with two columns, a header and horizontal lines above and below the table:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries \entryname &
   \bfseries \descriptionname\tabularnewline\hline}%
   \tabletail{\hline}
   \begin{supertabular}{|l|p{\glsdescwidth}|}%
   {\end{supertabular}}%
}

```

super3col The super3col style is like the super style, but with 3 columns:

```
\newglossarystyle{super3col}{%
```

Put the glossary in a supertabular environment with three columns and no head or tail:

```

\renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}}%
  {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
   {\end{supertabular}}%
}

```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

super3colborder The `super3colborder` style is like the `super3col` style, but with a border:

```
\newglossarystyle{super3colborder}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```
\newglossarystyle{super3colheader}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline}\tabletail{}}%
```

```

        \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
        {\end{supertabular}}%
    }

```

super3colheaderborder The `super3colheaderborder` style is like the `super3col` style but with a header and border:

```
\newglossarystyle{super3colheaderborder}{%
```

Base it on the `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{\glsdescwidth}|p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}

```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```
\newglossarystyle{super4col}{%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```

\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}}%
  \begin{supertabular}{|l|l|l|l|}%
  \end{supertabular}}%

```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```

\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%

```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```

\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
}

```

```

\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%

```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsnogroupskip
\renewcommand*\glsgroupskip{}%
\else
\renewcommand*\glsgroupskip}{& & & \tabularnewline}%
\fi
}

```

`super4colheader` The `super4colheader` style is like the `super4col` but with a header row.

```
\newglossarystyle{super4colheader}{%
```

Base it on the `glostylessuper4col` 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}{|l|l|l|l|}%
\end{supertabular}}%
}

```

`super4colborder` The `super4colborder` style is like the `super4col` but with a border.

```
\newglossarystyle{super4colborder}{%
```

Base it on the `glostylessuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|l|l|l|}%
\end{supertabular}}%
}

```

`super4colheaderborder` The `super4colheaderborder` style is like the `super4col` but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the `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 `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{ }\tabletail{ }%
  \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  \end{supertabular}}%
}
```

`altsuper4colheader` The `altsuper4colheader` style is like the `altsuper4col` but with a header row.

```
\newglossarystyle{altsuper4colheader}{%
```

Base it on the `glostylesuper4colheader` style:

```
\setglossarystyle{super4colheader}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline}\tabletail{ }%
  \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  \end{supertabular}}%
}
```

`altsuper4colborder` The `altsuper4colborder` style is like the `altsuper4col` but with a border.

```
\newglossarystyle{altsuper4colborder}{%
```

Base it on the `glostylesuper4colborder` style:

```
\setglossarystyle{super4colborder}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
```

```

    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}%
      {|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%
     {\end{supertabular}}%
  }

```

`altsuper4colheaderborder` The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```
\newglossarystyle{altsuper4colheaderborder}{%
```

Base it on the `glostylesuper4colheaderborder` style:

```
\setglossarystyle{super4colheaderborder}{%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

\renewenvironment{theglossary}%
  {\tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tablearnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}%
   {|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}

```

3.9 Glossary Styles using supertabular environment (`glossary-superragged.sty`)

The glossary styles defined in the `glossary-superragged` package use the `supertabular` environment. These styles are like those provided by the `glossary-super` package, except that the multiline columns have ragged right justification.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-superragged-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossary-superragged-2021-11-01.sty}
```

```
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2025/03/12 v4.57 (NLCT)]
```

Requires the `array` package:

```
\RequirePackage{array}
```

Requires the `supertabular` package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

`superragged` The superragged glossary style uses the supertabular environment.

```
\newglossarystyle{superragged}{%
```

Put the glossary in a supertabular environment with two columns and no head or tail:

```
\renewenvironment{theglossary}{%
  {\tablehead{} \tabletail{}}%
  \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
```

```

        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}

```

superraggedborder The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```

\newglossarystyle{superraggedborder}{%
Base it on the glostylesuperragged style:
    \setglossarystyle{superragged}%
Put the glossary in a supertabular environment with two columns and a horizontal line in the head and tail:
    \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
         \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
        {\end{supertabular}}%
}

```

superraggedheader The `superraggedheader` style is like the `super` style, but with a header:

```

\newglossarystyle{superraggedheader}{%
Base it on the glostylesuperragged style:
    \setglossarystyle{superragged}%
Put the glossary in a supertabular environment with two columns, a header and no tail:
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries \entryname & \bfseries \descriptionname
         \tabularnewline}%
         \tabletail{}}%
        {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
        {\end{supertabular}}%
}

```

superraggedheaderborder The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```

\newglossarystyle{superraggedheaderborder}{%
Base it on the glostylesuper style:
    \setglossarystyle{superragged}%
Put the glossary in a supertabular environment with two columns, a header and horizontal lines above and below the table:
    \renewenvironment{theglossary}%
        {\tablehead{\hline\bfseries \entryname &
         \bfseries \descriptionname\tabletabularnewline\hline}%
         \tabletail{\hline}%
         \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
        {\end{supertabular}}%
}

```

`superragged3col` The `superragged3col` style is like the `superragged` style, but with 3 columns:

```
\newglossarystyle{superragged3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}{%
  {\tablehead{ }\tabletail{ }}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

`superragged3colborder` The `superragged3colborder` style is like the `superragged3col` style, but with a border:

```
\newglossarystyle{superragged3colborder}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a supertabular environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
  >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
```

`superragged3colheader` The `superragged3colheader` style is like the `superragged3col` style but with a header row:

```
\newglossarystyle{superragged3colheader}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a supertabular environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline}\tabletail{}}%
  \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
  >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
```

`raggedright3colheaderborder` The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```
\newglossarystyle{superragged3colheaderborder}{%
```

Base it on the `glostylesuperragged3colborder` style:

```
\setglossarystyle{superragged3colborder}%
```

Put the glossary in a supertabular environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
  >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
```

`altsuperragged4col` The `altsuperragged4col` glossary style is like `altsuper4col` style in the `glossary-super` package but uses ragged right formatting in the description and page list columns.

```
\newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a supertabular environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}}%
  \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}1%
    >{\raggedright}p{\glspagelistwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*\glossaryheader{}
```

No group headings:

```
\renewcommand*\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

`altsuperragged4colheader` The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a supertabular environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
```

```

        {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline}\tabletail{}}%
        \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
        >{\raggedright}p{\glspagelistwidth}}}%
        {\end{supertabular}}}%
    }

```

altsuperragged4colborder The `altsuperragged4colborder` style is like the `altsuperragged4col` style but with a border.

```
\newglossarystyle{altsuperragged4colborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuper4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
    {||>{\raggedright}p{\glsdescwidth}|||%
    >{\raggedright}p{\glspagelistwidth}||}%
    {\end{supertabular}}%
}

```

altsuperragged4colheaderborder The `altsuperragged4colheaderborder` style is like the `altsuperragged4col` style but with a header and border.

```
\newglossarystyle{altsuperragged4colheaderborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
    {||>{\raggedright}p{\glsdescwidth}|||%
    >{\raggedright}p{\glspagelistwidth}||}%
    {\end{supertabular}}%
}

```

3.10 Tree Styles (`glossary-tree.sty`)

The `glossary-tree` style file defines glossary styles that have a tree-like structure. These are designed for hierarchical glossaries.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-tree-2020-03-19.sty}
```

```
\DeclareRelease{v4.49}{2021-11-01}{glossary-tree-2021-11-01.sty}
```

```
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossary-tree}[2025/03/12 v4.57 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%  
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax  
}
```

`\glstreenamefmt` Format used to display the name in the tree styles. (This may be counteracted by `\glsnamefont`.) This command was previously also used to format the group headings.

```
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```
\ifdef@idxitem  
{\newcommand{\glstreeitem}{\@idxitem}}  
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
```

`\glstreesubitem` Level 1 item used in index style.

```
\ifdef\subitem  
{\let\glstreesubitem\subitem}  
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
```

`\glstreesubsubitem` Level 1 item used in index style.

```
\ifdef\subsubitem  
{\let\glstreesubsubitem\subsubitem}  
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `almtree` style).

```
\newcommand{\glstreepredesc}{\space}
```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `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\glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%  
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%  
  \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%  
}%
```

Sub entries: level 1 entries use `\subitem`, levels greater than 1 use `\subsubitem`. The level (`##1`) shouldn't be 0, as that's catered by `\glossentry`, but for completeness, if the level is 0, `\item` is used. The name is put in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%  
  \ifcase##1\relax  
    % level 0  
    \item  
  \or  
    % level 1  
    \subitem  
    \glssubentryitem{##2}%  
  \else  
    % all other levels  
    \subsubitem
```

```

\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%

```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}}
```

indexgroup The `indexgroup` style is like the `index` style but has headings.

```
\newglossarystyle{indexgroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Add a heading for each group. This puts the group's title in bold followed by a vertical gap.

```

\renewcommand*\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\indexspace
}%
}

```

indexhypergroup The `indexhypergroup` style is like the `indexgroup` style but has hyper navigation.

```
\newglossarystyle{indexhypergroup}{%
```

Base it on the `glostyleindex` style:

```
\setglossarystyle{index}%
```

Put navigation links to the groups at the start of the glossary:

```

\renewcommand*\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%

```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```

\renewcommand*\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}

```

tree The `tree` glossary style is similar in style to the `index` style, but can have arbitrary levels.

```
\newglossarystyle{tree}{%
```

Set the paragraph indentation and skip:

```

\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%

```

Do nothing at the start of the `theglossary` environment:

```
\renewcommand*\glossaryheader}{%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}
```

Main (level 0) entries: name in bold, followed by symbol in brackets (if it exists), the description and the page list:

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}
```

treegroup Like the tree style but the glossary groups have headings.

```
\newglossarystyle{treegroup}{%
```

Base it on the `glostyletree` style:

```
\setglossarystyle{tree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
\renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
  \indexspace}%
}
```

treehypergroup The `treehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treehypergroup}{%
```

Base it on the `glostyletree` style:

```
\setglossarystyle{tree}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
  {\glslnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
  \indexspace}%
}
```

`\glstreeindent` Length governing left indent for each level of the tree style.

```
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
```

`treenoname` The `treenoname` glossary style is like the tree style, but doesn't print the name or symbol for sub-levels.

```
\newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
```

No header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries: the name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
```

Sub entries: level $\langle n \rangle$ is indented by $\langle n \rangle$ times `\glstreeindent`. The name and symbol are omitted. The description followed by the page list are displayed.

```
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

`treenonamegroup` Like the `treenoname` style but the glossary groups have headings.

```
\newglossarystyle{treenonamegroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Give each group a heading:

```
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgetgrouptitle{##1}}\par\indexspace}%
}
```

`treenamehypergroup` The `treenamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treenamehypergroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
```

`\glsfindwidesttoplevelname` Find the widest name over all parentless entries in the given glossary or glossaries.

```
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\foralllglossaries[#1]{\@gls@type}%
{%
\forglentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%
}%
\glsmeasurewidth{\dimen@}%
{\glstreenamefmt{\glstentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
}
```


Put the name to the left of the paragraph block.

```
\makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%  
  \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
\def\@gls@prevlevel{0}%  
}%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
\ifnum##1=1\relax  
  \glssubentryitem{##2}%  
\fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
\ifnum\@gls@prevlevel=##1\relax  
\else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
\@ifundefined{@glswidestname\romannumeral##1}{%  
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}}{%  
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%  
    \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
\ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
\setlength\glstreeindent\gls@tmplen  
\addtolength\glstreeindent\parindent  
\parindent\glstreeindent  
\else
```

Depth has decreased, so subtract width of the widest entry from the previous level to `\glstreeindent`. First determine the width of the widest entry for the previous level and store in `\glstreeindent`.

```
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%  
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%  
    \@glswidestname\space}}}{%  
  \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%  
    \csname @glswidestname\romannumeral\@gls@prevlevel  
    \endcsname\space}}}%
```

Subtract this length from the previous level's paragraph indent and set to `\glstreeindent`.

```

\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi

```

Set the hanging indentation.

```
\hangindent\glstreeindent
```

Put the name to the left of the paragraph block

```

\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%

```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```

\def\@gls@prevlevel{##1}%
}%

```

Vertical gap between groups is the same as that used by indices:

```

\renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}

```

`almtreegroup` Like the `almtree` style but the glossary groups have headings.

```
\newglossarystyle{almtreegroup}{%
```

Base it on the `glostylealmtree` style:

```
\setglossarystyle{almtree}%
```

Give each group a heading.

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}

```

`almtreehypergroup` The `almtreehypergroup` style is like the `almtreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{almtreehypergroup}{%
```

Base it on the `glostylealmtree` style:

```
\setglossarystyle{almtree}%
```

Put the navigation links in the header

```
\renewcommand*{\glossaryheader}{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}}
```

4 Multi-Lingual Support

4.1 Babel Captions (glossaries-babel.sty)

Many thanks to everyone who contributed to the translations both via email and on comp.text.tex. Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-babel-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-babel-2021-11-01.sty}
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Declare package

```
\ProvidesPackage{glossaries-babel}[2025/03/12 v4.57 (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{v4.57}{2025-03-12}

```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2025/03/12 v4.57 (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{v4.57}{2025-03-12}
```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2025/03/12 v4.57 (NLCT)]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

Load glossaries:

```
\RequirePackage{glossaries}
```

Add the new keys:

```
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
```

Add them to \@gls@keymap:

```
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}
```

Set the default values:

```
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
  \let\@glo@entryprefixfirstplural\@gls@default@value
}
```

Set the assignment code:

```
\appto\@newglossaryentryposthook{%
  \gls@assign@field{\@glo@label}{prefix}{\@glo@entryprefix}%
  \gls@assign@field{\@glo@label}{prefixplural}{\@glo@entryprefixplural}%
}
```

If `prefixfirst` has not been supplied, make it the same as `prefix`.

```
\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}%
  {\@glo@entryprefixfirst}%
```

If `prefixfirstplural` has not been supplied, make it the same as `prefixplural`.

```
\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @prefixplural\endcsname}{\@glo@label}%
  {prefixfirstplural}{\@glo@entryprefixfirstplural}%
}
```

Define commands to access these fields:

```
\glsentryprefixfirst
\newcommand*\glsentryprefixfirst[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
```

```
\glsentryprefixfirstplural
\newcommand*\glsentryprefixfirstplural[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
```

```
\glsentryprefix
\newcommand*\glsentryprefix[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
```

```
\glsentryprefixplural
\newcommand*\glsentryprefixplural[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
```

Now for the initial upper case variants:

```
\Glsentryprefixfirst
\newcommand*\Glsentryprefixfirst[1]{%
  \@Glsentryfield{#1}{prefixfirst}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefixfirst}{\Glsentryprefixfirst}
```

```
\Glsentryprefixfirstplural
\newcommand*\Glsentryprefixfirstplural[1]{%
  \@Glsentryfield{#1}{prefixfirstplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefixfirstplural}{\Glsentryprefixfirstplural}
```

```
\Glsentryprefix
\newcommand*\Glsentryprefix[1]{%
  \@Glsentryfield{#1}{prefix}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\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@glstetoklabel{#1}@prefix}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixplural`

```
\newcommand*{\ifglshasprefixplural}[3]{%
  \ifcempty{glo@glstetoklabel{#1}@prefixplural}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixfirst`

```
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@glstetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixfirstplural`

```
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glstetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
```

`\@glsprefix@record@hook` Need to take into account the possibility that glossaries-extra might be loaded with the record option.

```
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
  {\@glsxtr@record{#1}{#2}{glslink}}%
  {}%
}
```

`\glsprefixsep` Separator between prefix and term. Does nothing by default.

```
\newcommand{\glsprefixsep}{}
```

Define commands that insert the prefix before commands like `\gls`:

```
\pgls
```

```
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
```

```
\@pgls Unstarred version.
```

```
\newcommand*{\@pgls}[2][\%  
  \new@ifnextchar[  
  {\@pgls@{#1}{#2}}%  
  {\@pgls@{#1}{#2}[]}%  
  ]
```

```
\@pgls@ Read in the final optional argument:
```

```
\def\@pgls@#1#2[#3]{%  
  \@glsprefix@record@hook{#1}{#2}%  
  \glsdoifexists{#2}%  
  {%  
    \ifglsused{#2}%  
    {%  
      \ifglsshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%  
    }%  
    {%  
      \ifglsshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%  
    }%  
    \@gls@{#1}{#2}[#3]%  
  }%  
}
```

Similarly for the plural version:

```
\pglsp1
```

```
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
```

```
\@pglsp1 Unstarred version.
```

```
\newcommand*{\@pglsp1}[2][\%  
  \new@ifnextchar[  
  {\@pglsp1@{#1}{#2}}%  
  {\@pglsp1@{#1}{#2}[]}%  
  ]
```

```
\@pglsp1@ Read in the final optional argument:
```

```
\def\@pglsp1@#1#2[#3]{%  
  \@glsprefix@record@hook{#1}{#2}%  
  \glsdoifexists{#2}%  
  {%
```

```

\ifglused{#2}%
{%
  \ifglshasprefixplural{#2}\glentryprefixplural{#2}\glsprefixsep}{}%
}%
{%
  \ifglshasprefixfirstplural{#2}%
  {\glentryprefixfirstplural{#2}\glsprefixsep}{}%
}%
\@glsp1@{#1}{#2}[#3]%
}%
}

```

Now for the first letter upper case versions:

```

\@Pgl@s
\newrobustcmd{\@Pgl@s}{\@gl@s@hyp@opt\@Pgl@s}

```

Add sentence case mapping.

```

\glsmfuaddmap{\pgl@s}{\@Pgl@s}

```

\@Pgl@s Unstarred version.

```

\newcommand*{\@Pgl@s}[2][{}]{%
  \new@ifnextchar[%
  {\@Pgl@s@{#1}{#2}}{%
  {\@Pgl@s@{#1}{#2}[]}%
}

```

\@Pgl@s@ Read in the final optional argument:

```

\def\@Pgl@s@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {%
        \Glentryprefix{#2}%
        \glsprefixsep
        \@gl@s@{#1}{#2}[#3]%
      }%
      {\@Gl@s@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {%
        \Glentryprefixfirst{#2}%
        \glsprefixsep
        \@gl@s@{#1}{#2}[#3]%
      }%
    }%
  }%
}

```

```

        {\@Gls@{#1}{#2}[#3]}%
    }%
}

```

Similarly for the plural version:

```

\Pglspl
\newrobustcmd{\Pglspl}{\@gls@hyp@opt\@Pglspl}

```

Add sentence case mapping.

```

\glsmfuaddmap{\pglspl}{\Pglspl}

```

\@Pglspl Unstarred version.

```

\newcommand*{\@Pglspl}[2][ ]{%
  \new@ifnextchar[%
    {\@Pglspl@{#1}{#2}}%
    {\@Pglspl@{#1}{#2}[ ]}%
}

```

\@Pglspl@ Read in the final optional argument:

```

\def\@Pglspl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglschasprefixplural{#2}%
      {%
        \Glsentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
    {%
      \ifglschasprefixfirstplural{#2}%
      {%
        \Glsentryprefixfirstplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}

```

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]{\%
\@gls@prefix@record@hook{#1}{#2}\%
\glsdoifexists{#2}\%
{\%
\ifglsused{#2}\%
{\%
\ifgls@has@prefix{#2}\%
{\gls@supercase{\gls@entry@prefix{#2}\gls@prefixsep}}{\%
}\%
}\%
\ifgls@has@prefix@first{#2}\%
{\gls@supercase{\gls@entry@prefix@first{#2}\gls@prefixsep}}{\%
}\%
\@GLS@{#1}{#2}[#3]\%
}\%
}
```

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]{\%
\@gls@prefix@record@hook{#1}{#2}\%
\glsdoifexists{#2}\%
{\%
\ifglsused{#2}\%
{\%
\ifgls@has@prefix@plural{#2}\%
```

```

        {\glsuppercase{\glsentryprefixplural{#2}\glsprefixsep}}{}}%
    }%
    {%
    \ifglshasprefixfirstplural{#2}%
    {\glsuppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}}%
    }%
    \@GLSp1@{#1}{#2}{#3}%
}
}
}

```

6 Accessibility Support (glossaries-accsupp Code)

The glossaries-accsupp package is experimental. It is intended to provide a means of using the PDF accessibility support in glossary entries. See the accsupp documentation for further details about accessibility support.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-accsupp-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-accsupp-2021-11-01.sty}
\DeclareCurrentRelease{v4.57}{2025-03-12}

```

Package version number now in line with main glossaries package number.

```

\ProvidesPackage{glossaries-accsupp}[2025/03/12 v4.57 (NLCT)
Experimental glossaries accessibility]

```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

This package should be loaded before glossaries-extra, so complain if that has already been loaded.

```

\@ifpackageloaded{glossaries-extra}
{%

```

If the accsupp option was used, \@glsxtr@doaccsupp will have been set, otherwise it will be empty.

```

\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The ‘glossaries-accsupp’
package has been loaded\MessageBreak
after the ‘glossaries-extra’ package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the ‘accsupp’ option when you load\MessageBreak
‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
before loading ‘glossaries-extra’}%
\fi
}
{}

```

`\compatibleglossentry` Override style compatibility macros:

`\compatiblesubglossentry`

Required packages:

```
\RequirePackage{glossaries}
```

`\gls@accsupp@engine` There's currently only support for `accsupp`, but if you define `\gls@accsupp@engine` before loading `glossaries-accsupp`, you can prevent `accsupp` from being loaded. Redefining this command after `glossaries-accsupp` has loaded obviously won't do anything (and so is an internal command to deter casual use). If it is defined to something other than `accsupp` then `\gls@accessibility` will need to be defined to something appropriate.

```
\providecommand{\gls@accsupp@engine}{accsupp}
```

```
\gls@accessibility[<options>]{<PDF element>}{<value>}
{<content>}
```

`\gls@accessibility`

```
\providecommand{\gls@accessibility}[4][#4]
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
  }
}
{}
```

```
\glsaccessibility[<options>]{<PDF element>}{<value>}
{<content>}
```

`\glsaccessibility`

User-level command that includes debug info if required.

```
\newcommand{\glsaccessibility}[4][[]]{%
  \@glsshowaccsupp{#1}{#2}{#3}%
  \gls@accessibility{#1}{#2}{#3}{#4}%
}
```

6.1 Defining Replacement Text

The version 0.1 stored the replacement text in the symbol key. This has been changed to use the new keys defined here. Example of use:

```
%\newglossaryentry{dr}{name=Dr,description={},access={Doctor}}
%
```

access The replacement text corresponding to the name key:
`\define@key{glossentry}{access}{%
\def\@glo@access{#1}%
}`

textaccess The replacement text corresponding to the text key:
`\define@key{glossentry}{textaccess}{%
\def\@glo@textaccess{#1}%
}`

firstaccess The replacement text corresponding to the first key:
`\define@key{glossentry}{firstaccess}{%
\def\@glo@firstaccess{#1}%
}`

pluralaccess The replacement text corresponding to the plural key:
`\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}`

firstpluralaccess The replacement text corresponding to the firstplural key:
`\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}`

symbolaccess The replacement text corresponding to the symbol key:
`\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}`

symbolpluralaccess The replacement text corresponding to the symbolplural key:
`\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}`

descriptionaccess The replacement text corresponding to the description key:
`\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}`

descriptionpluralaccess The replacement text corresponding to the descriptionplural key:
`\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}`

shortaccess The replacement text corresponding to the short key:
`\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}`

`shortpluralaccess` The replacement text corresponding to the shortplural key:

```
\define@key{glossentry}{shortpluralaccess}{%
  \def\@glo@shortpluralaccess{#1}%
}
```

`longaccess` The replacement text corresponding to the long key:

```
\define@key{glossentry}{longaccess}{%
  \def\@glo@longaccess{#1}%
}
```

`longpluralaccess` The replacement text corresponding to the longplural key:

```
\define@key{glossentry}{longpluralaccess}{%
  \def\@glo@longpluralaccess{#1}%
}
```

There are now also keys that correspond to the user keys:

`user1access` The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%
  \def\@glo@useriaccess{#1}%
}
```

`user2access` The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%
  \def\@glo@useriiaccess{#1}%
}
```

`user3access` The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%
  \def\@glo@useriiiaccess{#1}%
}
```

`user4access` The replacement text corresponding to the user4 key:

```
\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
```

`user5access` The replacement text corresponding to the user5 key:

```
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the user6 key:

```
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%

  {descriptionaccess}{descaccess},%
  {descriptionpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{userviaccess},%
  {user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the access key was initialised to the value of the symbol key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for symbol but not for name so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
}
```

Initialise the other keys:

```
\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
```

```

\def@glo@longpluralaccess{\@glo@longaccess}%
\def@glo@useriaccess{\relax}%
\def@glo@useriiaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
\def@glo@useriiaaaccess{\relax}%
}

```

Add to the end hook:

```

\let@gls@oldnewglossaryentryposthook@newglossaryentryposthook
\renewcommand*{@newglossaryentryposthook}{%
@gls@oldnewglossaryentryposthook

```

Store the access information:

```

\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
@glo@longaccess}%

```

```

\expandafter
\protected@xdef\csname glo@\glo@label @longpluralaccess\endcsname{%
  \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriaccess\endcsname{%
  \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiaccess\endcsname{%
  \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}

```

6.2 Accessing Replacement Text

`\glsentryaccess` Get the value of the access key for the entry with the given label:

```

\newcommand*{\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}

```

`\glsentrytextaccess` Get the value of the textaccess key for the entry with the given label:

```

\newcommand*{\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}

```

`\glsentryfirstaccess` Get the value of the firstaccess key for the entry with the given label:

```

\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}

```

`\glsentrypluralaccess` Get the value of the pluralaccess key for the entry with the given label:

```

\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}

```

`\glsentryfirstpluralaccess` Get the value of the firstpluralaccess key for the entry with the given label:

```

\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}

```

`\glsentrysymbolaccess` Get the value of the `symbolaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
```

`\glsentrysymbolpluralaccess` Get the value of the `symbolpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
```

`\glsentrydescaccess` Get the value of the `descriptionaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
```

`\glsentrydescpluralaccess` Get the value of the `descriptionpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
```

`\glsentryshortaccess` Get the value of the `shortaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
```

`\glsentryshortpluralaccess` Get the value of the `shortpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
```

`\glsentrylongaccess` Get the value of the `longaccess` key for the entry with the given label:

```
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
```

`\glsentrylongpluralaccess` Get the value of the `longpluralaccess` key for the entry with the given label:

```
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
```

`\glsentryuseriaccess` Get the value of the `user1access` key for the entry with the given label:

```
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
```

`\glsentryuseriiaccess` Get the value of the `user2access` key for the entry with the given label:

```
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
```

`\glsentryuseriiiaccess` Get the value of the `user3access` key for the entry with the given label:

```
\newcommand*\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
```

`\glsentryuserivaccess` Get the value of the `user4access` key for the entry with the given label:

```
\newcommand*\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
```

`\glsentryuservaccess` Get the value of the `user5access` key for the entry with the given label:

```
\newcommand*\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
```

`\glsentryuserviaccess` Get the value of the `user6access` key for the entry with the given label:

```
\newcommand*\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
```

There are three types of replacement text:

Alt Description of some content that’s non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require **E**, the `symbol` key would require **ActualText**, and a field that contains an image would require **Alt**.

```
\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}
```

`\glsfieldaccsupp`

Test if there’s a command called `\gls<field>accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If `glossaries-extra` has been loaded, this first checks for `\glsxtr<category><field>accsupp` and `\glsxtr<category>accsupp`.

```
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
```

```

\ifcsdef{glxstr\glscategory{#4}#3accsupp}%
{\csname glxstr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
{%
\ifcsdef{glxstr\glscategory{#4}accsupp}%
{\csname glxstr\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}}%
}%
}

```

```
\xglsfielddaccsupp{<replacement>}{<content>}{<field>}
{<label>}
```

\xglsfielddaccsupp

As \glsfielddaccsupp but fully expand replacement text.

```

\newcommand{\xglsfielddaccsupp}[1]{%
\protected@edef\@gls@replacementtext{#1}%
\expandafter\glsfielddaccsupp\expandafter{\@gls@replacementtext}%
}

```

```
\glsshortaccsupp{<replacement text>}{<text>}
```

\glsshortaccsupp

```
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

```
\glsshortplaccsupp{<replacement text>}{<text>}
```

\glsshortplaccsupp

```
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

```
\glsaccsupp{<replacement text>}{<text>}
```

\glsaccsupp

```
\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{<label>}{<field>}{<content>}
{<replacement>}

```

`\@gls@fieldaccess@display`

```

\newcommand*{\@gls@fieldaccess@display}[4]{%
\protected@edef\@glo@access{#4}%
\ifdequal\@glo@access\@gls@noaccess
{#3}%
\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}

```

`\glsnameaccessdisplay` Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```

\newrobustcmd*{\glsnameaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@access}%
{#1}%
{\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}

```

`\glstextaccessdisplay` As above but for the textaccess replacement text.

```

\newrobustcmd*{\glstextaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@textaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}

```

`\glspluralaccessdisplay` As above but for the pluralaccess replacement text.

```

\newrobustcmd*{\glspluralaccessdisplay}[2]{%
\ifcsundef{glo@glsdetoklabel{#2}@pluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}

```

`\glsfirstaccessdisplay` As above but for the firstaccess replacement text.

```

\newrobustcmd*\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@firstaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}

glsfirstpluralaccessdisplay As above but for the firstpluralaccess replacement text.
\newrobustcmd*\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@firstpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}

\glsymbolaccessdisplay As above but for the symbolaccess replacement text.
\newrobustcmd*\glsymbolaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@symbolaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}

glsymbolpluralaccessdisplay As above but for the symbolpluralaccess replacement text.
\newrobustcmd*\glsymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@symbolpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}

glsdescriptionaccessdisplay As above but for the descriptionaccess replacement text.
\newrobustcmd*\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}

descriptionpluralaccessdisplay As above but for the descriptionpluralaccess replacement text.
\newrobustcmd*\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}

\glsshortaccessdisplay As above but for the shortaccess replacement text.
\newrobustcmd*\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}

```

`\glsshortpluralaccessdisplay` As above but for the shortpluralaccess replacement text.

```

\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstryshortpluralaccess{#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}{\glstrylongaccess{#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}{\glstrylongpluralaccess{#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}{\glstryuseriaccess{#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}{\glstryuseriiaccess{#2}}}%
}

```

`\glsuseriiiaccessdisplay` As above but for the user3access replacement text.

```

\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glstryuseriiiaccess{#2}}}%
}

```

`\glsuserivaccessdisplay` As above but for the user4access replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glstryuserivaccess{#2}}}%
}

```

```

\glsuservaccessdisplay As above but for the user5access replacement text.
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}

\glsuserviaccessdisplay As above but for the user6access replacement text.
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{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 infor-
mation 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
  {\glsupercase{\glsentryplural{\glslabel}}}%
  {\glslabel}%
\glsupercase{\glsinsert}%
}%
{%
```

First use

```
\glsfirstpluralacessdisplay
  {\glsupercase{\glsentryfirstplural{\glslabel}}}%
  {\glslabel}%
\glsupercase{\glsinsert}%
}%
}%
}%
{%
```

Singular form

```
\glscapscale
{%
```

Don't adjust case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
{%
```

First use

```
\glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
{%
```

First use

```
\glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglsused\glslabel  
{%
```

Subsequent use

```
\glstextaccessdisplay  
{\glsuppercase{\glsentrytext{\glslabel}}}{\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
{%
```

First use

```
\glsfirstaccessdisplay  
{\glsuppercase{\glsentryfirst{\glslabel}}}{\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
}%  
}%
```

```

    {%
Custom text provided in \glsdisp. (The insert should be empty at this point.)
The accessibility information, if required, will have to be explicitly included in
the custom text.
    \glscustomtext\glsinsert
    }%
}

```

`\glsngenacfmt` Redefine to include accessibility information.

```

\renewcommand*{\glsngenacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%

```

Subsequent use:

```

\glsifplural
{%

```

Subsequent plural form:

```

\glscapscase
{%

```

Subsequent plural form, don't adjust case:

```

\acronymfont
{\glsshortpluralaccessdisplay
{\glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%

```

Subsequent plural form, make first letter upper case:

```

\acronymfont
{\glsshortpluralaccessdisplay
{\Glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%

```

Subsequent plural form, all caps:

```

\glsuppercase
{\acronymfont
{\glsshortpluralaccessdisplay
{\glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
{%

```

Subsequent singular form

```

\glscapscase
{%

```

Subsequent singular form, don't adjust case:

```
\acronymfont
{\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent singular form, make first letter upper case:

```
\acronymfont
{\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent singular form, all caps:

```
\glsuppercase
{\acronymfont{%
\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
{%
```

First use:

```
\glsifplural
{%
```

First use plural form:

```
\glscapscase
{%
```

First use plural form, don't adjust case:

```
\genplacrfullformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, make first letter upper case:

```
\Genplacrfullformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, all caps:

```
\glsuppercase
{\genplacrfullformat{\glslabel}{\glsinsert}}%
}%
}%
{%
```

First use singular form

```
\glscapscase
{%
```

First use singular form, don't adjust case:

```
\genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, make first letter upper case:

```
\Genacrfullformat{\glslabel}{\glsinsert}%  
}%  
{%
```

First use singular form, all caps:

```
\glsuppercase  
{\genacrfullformat{\glslabel}{\glsinsert}}%  
}%  
}%  
}%  
}%  
{%
```

User supplied text. (The insert should be empty at this point.) The accessibility information, if required, will have to be explicitly included in the custom text.

```
\glscustomtext  
}%  
}
```

`\genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\genacrfullformat}[2]{%  
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space  
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%  
}
```

`\Genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\Genacrfullformat}[2]{%  
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space  
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%  
}
```

`\genplacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\genplacrfullformat}[2]{%  
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space  
  (\glsshortpluralaccessdisplay  
  {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%  
}
```

`\Genplacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\Genplacrfullformat}[2]{%  
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space  
  (\glsshortpluralaccessdisplay  
  {\protect\firstacronymfont{\Glsentryshortpl{#1}}}{#1})%  
}
```

```

\@acrshort
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
  }%
}
Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@Acrshort
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
  }%
}
Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}

\@ACRshort
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay
        {\glsupercase{\glsentryshort{#2}}}{#2}}#3%
    }%
  }%
}

```

```

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@acrlong
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
  }

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@Acrlong
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glsapscase\@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\glsifcaps\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{%
    \glsuppercase{\glsentrylong{#2}}{#2}#3}%
  }%
}

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
}%

\gls@postlinkhook
}

\@gls@text@
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}{#2}#3}%
}

\@Gls@text@
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\Glsentrytext{#2}}{#2}#3}%
}

\@GLS@text@
\def\@GLS@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    \gls@text@accessdisplay{\glsuppercase{\glsentrytext{#2}}{#2}}{#2}%
    \glsuppercase{#3}}%
}

\@gls@first@
\def\@gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\glsentryfirst{#2}}{#2}#3}%
}

\@Gls@first@
\def\@Gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}

\@GLS@first@
\def\@GLS@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    \gls@first@accessdisplay{\glsuppercase{\glsentryfirst{#2}}{#2}}{#2}%
    \glsuppercase{#3}}%
}

```

```

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\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{\glsupercase{\glsentryname{#2}}}{#2}%
    \glsupercase{#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}}%
  }
}

\@glsymbol@
\def\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSsymbol@
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsymbolaccessdisplay{\glsuppercase{\glsentrysymbol{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

```

\@glssymbolplural@
\def\@glssymbolplural@#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{\Glsentrysymbolplural{#2}}{#2}#3}%
}

\@GLSsymbolplural@
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolpluralaccessdisplay{\glsupercase{\glsentrysymbolplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseri@
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}

\@Glsuseri@
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}

\@GLSuseri@
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\glsupercase{\glsentryuseri{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuserii@#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{\glsuppercase{\glsentryuseriii{#2}}{#2}%
  \glsuppercase{#3}}}%
}

\@glsuseriv@
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}

\@Glsuseriv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}

\@GLSuseriv@
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\glsuppercase{\glsentryuseriv{#2}}{#2}%
  \glsuppercase{#3}}}%
}

\@glsuserv@
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}

\@Glsuserv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

\@GLSuserv@
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuservaccessdisplay{\glsuppercase{\glsentryuserv{#2}}{#2}%
  \glsuppercase{#3}}}%
}

```

```

\@glsuservi@
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}

\@Glsuservi@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}

\@GLSuservi@
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    \glsuserviaccessdisplay{\glsuppercase{\glsentryuservi{#2}}}{#2}%
    \glsuppercase{#3}}%
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%

```

```

        \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
    }%
}
\renewcommand*\glossentrysymbol[1]{%
    \glsdoifexists{#1}%
    {%
        \glsymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
    }%
}
\renewcommand*\Glossentrysymbol[1]{%
    \glsdoifexists{#1}%
    {%
        \glsymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
    }%
}

```

\accsuppglossaryentryfield

```

\newcommand*\accsuppglossaryentryfield[5]{%
    \glossaryentryfield{#1}%
    {\glsnameaccessdisplay{#2}{#1}}%
    {\glsdescriptionaccessdisplay{#3}{#1}}%
    {\glsymbolaccessdisplay{#4}{#1}}{#5}%
}

```

accsuppglossarysubentryfield

```

\newcommand*\accsuppglossarysubentryfield[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glsnameaccessdisplay{#3}{#2}}%
    {\glsdescriptionaccessdisplay{#4}{#2}}%
    {\glsymbolaccessdisplay{#5}{#2}}{#6}%
}

```

6.4 Acronyms

Redefine acronym styles provided by glossaries:

`long-short` *<long>* (*<short>*) acronym style.

```

\renewacronymstyle{long-short}%
{%

```

Check for long form in case this is a mixed glossary.

```

    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat[2]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
        {\protect\firstacronymfont{\glsentryshort{##1}}{##1})%

```

```

}%
\renewcommand*\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
  (\glsshortaccessdisplay
    {\protect\firstacronymfont{\Glsentryshort{##1}}{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}{##1}}
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\Glspluralsuffix}%
}

```

short-long *<short>* (*<long>*) acronym style.

```

\renewacronymstyle{short-long}%
{%

```

Check for long form in case this is a mixed glossary.

```

  \ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\Glsentryshort{##1}}{##1}##2\space
      (\glslongaccessdisplay{\Glsentrylong{##1}}{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\Glsentryshort{##1}}{##1}##2\space
      (\glslongaccessdisplay{\Glsentrylong{##1}}{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\Glsentryshortpl{##1}}{##1}##2\space
      (\glslongpluralaccessdisplay
        {\Glsentrylongpl{##1}}{##1}})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%

```

```

\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`long-short-desc` *<long>* (*{(short)}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1})}%
}

```

`long-sc-short-desc` *<long>* (`\textsc{short}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1})}%
}

```

`long-sm-short-desc` *<long>* (`\textsmaller{short}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%

```

```

\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

short-long-desc *<short>* (*{<long>}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

sc-short-long-desc *<long>* (*\textsc{<short>}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

sm-short-long-desc *<long>* (*\textsmaller{<short>}*) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%

```

```

\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

dua *<long>* only acronym style.

```

\renewacronymstyle{dua}%
{

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{
\ifglsahaslong{\glslabel}%
{
\glsifplural
{

```

Plural form:

```

\glscapscase
{

```

Plural form, don't adjust case:

```

\glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
\glsinsert
}%
{

```

Plural form, make first letter upper case:

```

\glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
\glsinsert
}%
{

```

Plural form, all caps:

```

\glslongpluralaccessdisplay
{\glsuppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
\glsuppercase{\glsinsert}%
}%
{

```

Singular form

```

\glscapscase
{

```

Singular form, don't adjust case:

```

\glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
}%
{

```

Subsequent singular form, make first letter upper case:

```

\glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
}%
{

```

Subsequent singular form, all caps:

```

\glslongaccessdisplay
  {\glsuppercase
   {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
\glsuppercase{\glsinsert}%
  }%
}%
}%
{%

```

Not an acronym:

```

\glsgenentryfmt
}%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\glsuppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsuppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%

```

```

\glslongaccessdisplay{\glsentrylong{##1}}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*\Glsentryfull}[1]{%
\glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*\Glsentryfullpl}[1]{%
\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\Glsentryfullpl}[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*\acronymsort}[2]{##2}%
}%

```

footnote *<short>*\footnote{*<long>*} acronym style.

```

\renewacronymstyle{footnote}%
{%
Check for long form in case this is a mixed glossary.
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%

```

Need to ensure hyperlinks are switched off on first use:

```

\glshyperfirstfalse
\renewcommand*\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%

```

```

\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%

```

Don't use footnotes for \acrfull:

```

\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
(\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
(\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay
{\glsupercase
{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
(\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
{\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortpluralaccessdisplay
{\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%

```

```

\glsshortpluralaccessdisplay
  {\glsupercase
   {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}%

```

Similarly for `\glsentryfull` etc:

```

\renewcommand*{\glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
  {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
  (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
}

```

`footnote-sc` `\textsc{<short>}\footnote{<long>}` acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%

```

`footnote-sm` `\textsmaller{<short>}\footnote{<long>}` acronym style.

```

\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%

```

`footnote-desc` `<short>\footnote{<long>}` acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

```

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }

```

```
\glsdefaultshortaccess{<long>}{<short>}
```

\glsdefaultshortaccess

Default shortaccess value.

```
\newcommand*{\glsdefaultshortaccess}[2]{##1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*{\newacronymhook}{%
  \protected@edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
```

`\DefaultNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionFootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\SmallNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

The following are kept for compatibility with versions before 3.0:

`\glsshortaccesskey`

```
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
```

`\glsshortpluralaccesskey`

```
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
```

`\glslongaccesskey`

```
\newcommand*{\glslongaccesskey}{\glslongkey access}%
```

`\glslongpluralaccesskey`

```
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

6.5 Debugging Commands

`\showglongnameaccess`

```
\newcommand*{\showglongnameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
```

`\showglotextaccess`

```
\newcommand*{\showglotextaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
```

```

\showglopluralaccess
\newcommand*{\showglopluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@pluralaccess\endcsname
}

```

```

\showglofirstaccess
\newcommand*{\showglofirstaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@firstaccess\endcsname
}

```

```

\showglofirstpluralaccess
\newcommand*{\showglofirstpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@firstpluralaccess\endcsname
}

```

```

\showglosymbolaccess
\newcommand*{\showglosymbolaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolaccess\endcsname
}

```

```

\showglosymbolpluralaccess
\newcommand*{\showglosymbolpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@symbolpluralaccess\endcsname
}

```

```

\showglodescaccess
\newcommand*{\showglodescaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#1}@descaccess\endcsname
}

```

```

\showglodescpluralaccess
\newcommand*{\showglodescpluralaccess}[1]{%
\expandafter\show\csname glo@glstetoklabel{#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@glsdetoklabel{#1}@longpluralaccess\endcsname  
}
```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original `glossaries` xindy and `makeindex` formatting, so can be used with old documents that had customized style files, but hyperlinks may not work properly.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-207-2020-03-19.sty}  
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-207-2021-11-01.sty}  
\DeclareCurrentRelease{v4.57}{2025-03-12}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2025/03/12 v4.57 (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{v4.57}{2025-03-12}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2025/03/12 v4.57 (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\gls@showtarget\@gobble
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
  
```

```

\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
  \nfss@text{\glsshowtargetfont [#1]}%
\else
  \ifinner
    {\glsshowtargetfont [#1]}%
  \else
    \glsshowtargetouter{#1}%
  \fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{}
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp

```

```

\expandafter\glsshowtarget\expandafter{\@gls@tmp}%
\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error, warn, ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part, chapter, section,%

```

```

subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}
\newcommand*{\@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
  \ifcase\gls@numberedsection@nr\relax
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{%
      \label{\glsautoprefix@glo@type}}%
  \or
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix@glo@type}}%
  \fi
}
\@ifpackageloaded{classicthesis}
{
\newcommand*{\@glossary@default@style}{index}
\newcommand*{\@glossary@default@style}{list}
\define@key{glossaries.sty}{style}{%
  \def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
  \DeclareOptionX{#1}{#2}%
  \DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{}
\renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*\@glo@seeautonumberlist{}%
  \def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glsesclocationstrue
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}

```

```

\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}}
\IfFileExists{supertabular.sty}{%
  \newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}[=v4.46]}{%
  \newcommand*{\@gls@loadsuper}{}}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}}
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}[=v4.46]}
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
  \ifdefstring{\@glossary@default@style}{list}%
  {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
  \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  {}%
\fi

```

```

}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theH@glossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@glsdefaultsort
      \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandardsort{\@glo@sort}{##1}{##2}%
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%

```

```

\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sort@allowed\@gobble
}
\newcommand*{\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\edef\@glo@type{\csname glo@##1@type\endcsname}%
\edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
}

```

```

    \fi
  }%
  \let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@none}{%
  \def\do@glo@storeentry##1{%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
  }%
  \def\@gls@setsort##1{%
  \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
  {Option sort=none not allowed with \string##1}%
  {(Use sort=def instead)}}%
}
\newcommand*\glsdefmain}{%
  \if@gls@docloaded
    \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
  \else
    \newglossary{main}{gls}{glo}{\glossaryname}%
  \fi
  \newcommand*\gls@tr@set@main@toctitle}{%
    \translatelet{\glossarytoctitle}{Glossary}%
  }%
}
\newcommand*\glsdefaulttype}{main}
\newcommand*\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
  \let\glsdefaulttype\relax
  \renewcommand*\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
  \ifglsacronym
    \renewcommand*\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*\acronymtype}{acronym}%
      \newcommand*\gls@tr@set@acronym@toctitle}{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%
  \ifglsacronym
    \ifbool{glscompatible-3.07}{%
      }%
    }%

```

```

        \providecommand*\printacronyms}[1] [] {%
            \printglossary[type=\acronymtype,#1]}%
    }%
\fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
    \glsacronymtrue
    \def\@gls@do@acronymsdef{%
        \DeclareAcronymList{acronym}%
        \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
        \renewcommand*\@acronymtype}{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
}
\newcommand*\@gls@acronymlists{}
\newcommand*\@addtoacronymlists}[1]{%
    \ifx\@gls@acronymlists\@empty
        \protected@xdef\@gls@acronymlists{#1}%
    \else
        \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
    \fi
}
\newcommand*\DeclareAcronymList}[1]{%
    \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
    \edef\@do@gls@islistofacronyms{%
        \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
    \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
    \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
        \def\@gls@before{##1}\def\@gls@after{##2}}%
    \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
    \ifx\@gls@after\@nnil
        #4%
    \else
        #3%
    \fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
    \glsIfListOfAcronyms{#1}%
    {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\SetAcronymLists}[1]{%
    \renewcommand*\@gls@acronymlists{#1}%

```

```

}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

}
\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%

```

```

\def\@glo@sort{#1#2}%
\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefvoid\@glo@sort
{}%
{%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent\@secondoftwo
\let\@text@composite\@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

```

```

        \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]{%
\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
\forallglslentries[\@gls@types,\@ignored@glossaries]{\@glslentry}%
{\ifdefempty{\@glslentry}{\immediate\write\@glsl@labelsfile{\@glslentry}}}%
\immediate\closeout\@glsl@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
[\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
\ifnum\gls@automake@nr=1\relax
\glsautomakefalse
\else
\glsautomaketrue
\fi
\ifglsautomake
\renewcommand*\@gls@doautomake}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with automake=true}
{%
Either remove the automake=true setting or
add \string\makeglossaries\space to your document preamble.%
}%
}%
\else
\renewcommand*\@gls@doautomake}{}%
\fi
}
\glsautomakefalse

```

```

\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifglssavewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
    {%
      \booltrue{glscompatible-3.07}%
    }%
  }%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
instead. If you \MessageBreak need the original definition of
\string\makeglossary\space use \MessageBreak the package options
kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
  \def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%

```

```

{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
\newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]}%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{}
\newcommand*{\@gls@numbersdef}{%
\newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
\newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]}%
\newcommand*{\gls@tr@set@numbers@toctitle}{%
\translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
}%
}%
\@gls@declareoption{index}{%
\ifx\@gls@do@indexdef\@empty
\let\@gls@do@indexdef\@gls@indexdef
\fi
}
}

```

```

\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]%
  \newcommand*{\newterm}[2][\%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \@expandtwoargs
  \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
  \ifin@
  \@use@ption
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.46]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
  \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
  \def\@gls@setupshortcuts{%
    \ifglsacrshortcuts
    \DefineAcronymSynonyms
  }
  \fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsentrycounter
  \let\@gls@doentrycounterdef\relax
\else
  \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi

```

```

\ifglssubentrycounter
  \let\@gls@dosingentrycounterdef\relax
\else
  \let\@gls@dosingentrycounterdef\@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@dosingentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
  \ifcsundef{chapter}{}%
  {%
    \let\@gls@old@chapter\@chapter
    \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
    \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
  }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{}
\newcommand*{\@onlypremakeg}[1]{%
  \ifx\@gls@onlypremakeg\@empty
    \def\@gls@onlypremakeg{#1}%
  \else
    \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
    \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
  \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
  \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
  used before \string\makeglossaries}{You can't use
  \string#1\space after \string\makeglossaries}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
  \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}

```

```

\providecommand*\pagelistname}{Page List}
\providecommand*\glssymbolsgroupname}{Symbols}
\providecommand*\glsnumbersgroupname}{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp
            \renewcommand*\glossaryname}{\translate{Glossary}}}%
          }%
          \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
        }%
      }%
    }%
  }%
  }%
  {}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
  }%
}

```

```

    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
      {%
        \csuse{gls@tr@set@#1@toctitle}%
      }%
      {%
        \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
      }%
    }%
  \renewcommand*\glossaryname{\translate{Glossary}}%
  \renewcommand*\acronymname{\translate{Acronyms}}%
  \renewcommand*\entryname{\translate{Notation (glossaries)}}%
  \renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
  \renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
  \renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
  \renewcommand*\glsymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
}{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
  \renewcommand*\gls@istfilebase{#1}%
  \ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
}

```

```

\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*{\gls@istfilebase}{\jobname}
\newcommand*{@\istfilename}[1]{%
\newcommand*{\glscompositor}{.}
\newcommand*{\glsSetCompositor}[1]{%
  \renewcommand*{\glscompositor}{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*{@\glsAlphacompositor}{\glscompositor}
\ifglxindy
  \newcommand*{\glsSetAlphaCompositor}[1]{%
    \renewcommand*{@\glsAlphacompositor}{#1}}
\else
  \newcommand*{\glsSetAlphaCompositor}[1]{%
    \glsnoindywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{%
\newcommand*{\glsSetSuffixF}[1]{%
  \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{%
\newcommand*{\glsSetSuffixFF}[1]{%
  \renewcommand*{\gls@suffixFF}{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*{\glsnumberformat}[1]{\glshypernumber{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{%
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%

```

```

{%
  \@glossarysection{#1}{#2}%
}%
{%
  \@pglossarysection{#1}{#2}%
}%
\glsglossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\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*\gls@toc}[2]{%
  \ifglstoc
  \ifglsnumberline
  \addcontentsline{toc}{#2}{\protect\numberline{}#1}%
  \else
  \addcontentsline{toc}{#2}{#1}%
  \fi
\fi
}
\newcommand*\glsnoxywarning}[1]{%

```

```

    \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsmakeindexwarning}[1]{%
    \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglxindy
    \edef\xdyattributes{\string"default\string"}%
\fi
\ifglxindy
    \edef\xdyattributelist{}%
\fi
\ifglxindy
    \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
    \def\do@ifinlist##1,#1,##2\end@ifinlist{%
        \def@gls@listsuffix{##2}%
        \ifx@gls@listsuffix\@empty
            #4%
        \else
            #3%
        \fi
    }%
    \do@ifinlist,#2,#1,\end@ifinlist
}
\ifglxindy
    \newcommand*\@xdycounters{\glscounter}
    \newcommand*\GlsAddXdyCounters[1]{%
        \for@gls@ctr:=#1\do{%
            \edef\do@addcounter{%
                \noexpand@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
                {%
                    \noexpand\edef\noexpand\xdycounters{\@xdycounters,%
                        \noexpand@gls@ctr}%
                }%
            }%
            \do@addcounter
        }
    }
    \@onlypremakeg\GlsAddXdyCounters
\else
    \newcommand*\GlsAddXdyCounters[1]{%
        \glsnoinxindywarning\GlsAddXdyAttribute
    }
\fi
\newcommand*\@disabled@glssaddxdycounters{%
    \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}

```

```

}
\ifglxsindy
\newcommand*\glsaddxdyattribute[2]{%
\edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
\ifx\@xdyattributelist\@empty
\edef\@xdyattributelist{#1}%
\else
\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi
\@for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxdyattribute{%
\noexpand\glsaddxdyattribute{#1}{\@this@counter}%
}
\gls@do@addxdyattribute
}%
\let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxindywarning\GlsAddXdyAttribute}
\fi
\ifglxsindy
\newcommand*\@gls@addpredefinedattributes{%
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}

```

```

\GlsAddXdyAttribute{hyperarm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*\@GlsAddXdyAlphabet}[2]{%
  \edef\@xdyuseralphabets{%
    \@xdyuseralphabets ^^J
    (define-alphabet "#1" (#2))}}
\else
  \newcommand*\@GlsAddXdyAlphabet}[2]{%
    \glsnoxindywarning\@GlsAddXdyAlphabet}
\fi
\ifglxindy
  \newcommand*\@gls@xdy@locationlist{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \expandafter
  \edef\cename @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\cename @gls@xdy@Lclass@roman-page-numbers\endcsname{
    :sep \string"\@gls@roman\string"%
  }%
\fi

```

```

    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][ ]{%
  \def\@gls@tmp{#1}%
  \ifx\@gls@tmp\@empty
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glsopenbrace\string" #3
          :sep \string"\glsclosebrace\string"))
    }%
  \else
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#2\string"^^J\space\space
        \space(:sep "\glsopenbrace"
          #1
          :sep "\glsclosebrace\glsopenbrace" #3
          :sep "\glsclosebrace"))
    }%
  \fi
\fi

```

```

\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation}[2]{%
    \glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
  \string"roman-page-numbers\string"^^J\space\space\space
  \string"arabic-page-numbers\string"^^J\space\space\space
  \string"arabic-section-numbers\string"^^J\space\space\space
  \string"alpha-page-numbers\string"^^J\space\space\space
  \string"Roman-page-numbers\string"^^J\space\space\space
  \string"Alpha-page-numbers\string"^^J\space\space\space
  \string"Appendix-page-numbers\string"
  \@xdyuserlocationnames^^J\space\space\space
  \string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
  \def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
  \glsnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\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]{%
  \glsnoxindywarning\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
\ifglxindy
  \ifgls@xindy@glsnumbers
    \def\xdylettergroups{(define-letter-group
      \string"glslnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\xdylettergroups{}
    \fi
  \fi
\fi

```

```

\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\glsdefaulttype]{%
  \edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forallglsentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{glo@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*\ifglsused[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already

```

```

        been defined}{#2}%
    }
\newcommand{\glsdoifexistsorwarn}[2]{%
    \ifglsentryexists{#1}{#2}{%
        \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
            has not been defined}%
    }%
}
\newcommand{\glsdoifexistsordo}[3]{%
    \ifglsentryexists{#1}{#2}{%
        \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
            has not been defined}{You need to define a glossary entry before you
            can use it.}%
        #3%
    }%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
    \ifglossaryexists*{#1}%
    {%
        \PackageError{glossaries}{Glossary type ‘#1’ already exists}{%
            #3%
        }%
    }%
    {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
    \glsdoifexists{#1}%
    {%
        \def\do@glshaschildren{#3}%
        \edef\@gls@thislabel{\glsdetoklabel{#1}}%
        \expandafter\for\glsentries\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]{%
  \ifcempty{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}\@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@forbidtext}[1]{%
  \ifboolexpr{test {\ifdefstring{#1}{tex}}

```

```

        or test {\ifdefstring{#1}{TEX}}
    {%
      \def#1{nottex}%
      \PackageError{glossaries}%
        {Forbidden '.tex' extension replaced with '.nottex'}%
        {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
        Don't use '.tex' as an extension for a temporary file.}%
    }%
  }%
}
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }{}%
    \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
    \expandafter\gdef\csname glolist@#2\endcsname{,}%
    \expandafter\edef\csname @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 @gls@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
\csname @gls@#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{gllist@#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\@gls@name{#1}%
}
\define@key{glossentry}{description}{%

```

```

\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*{\@gls@default@value}{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
\renewcommand*{\gls@xr@key}{#1}%
\gls@checkseeallowed
\def#2{#3}%
\@glo@seeautonumberlist

```

```

}
\newcommand*\gls@xr@key}{see}
\newcommand*\gls@checkseeallowed}{%
  \gls@see@noindex
}
\newcommand*\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning{glossaries}%
  {'\gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
  [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
}{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \gls@savenonumberlist{false}%
  \fi
}
\newcommand*\@gls@savenonumberlist}[1]{}
\newcommand*\@gls@initnonumberlist}{%
\newcommand*\@gls@storenonumberlist}[1]{}
\newcommand*\@gls@enablesavenonumberlist}{%
  \renewcommand*\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet\glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
    }%
    {}%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
}

```

```

\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@glsdefaultplural{}
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \iflssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
  Maybe makeglossaries + rerun required}%
\else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
}

```

```

    }%
  \fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
}
\ifdefequal{#4}{\@gls@default@value}%
  {%
    \edef\@gls@value{\expandonce{#1}}%
    \@gls@noexpand@field{#2}{#3}{\@gls@value}%
  }%
  {%
    \@gls@noexpand@field{#2}{#3}{#4}%
  }%
}
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
  \protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
      {%
        \@gls@expand@field{#2}{#3}{#4}%
      }%
    }%
  }%
}

```

```

        \csuse{gls@assign@#3@field}{#2}{#4}%
      }%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \@@gls@expand@field{#2}{#3}{#1}%
    }%
    {%
      \@@gls@expand@field{#2}{#3}{#4}%
    }%
  }%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\gls@defdocnewglossaryentry{%
  \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
}

```

```

{}%
\ifglsentryexists{#1}{}%
{%
  \gls@defglossaryentry{#1}{#2}%
}%
\@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{-}{-}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglsentryexists{#1}{}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar^^J%
    \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
        \glspercentchar
      }%
    }%
  }%
  {}%
}%
\gls@writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar%
}%
}
\newcommand*\@gls@keymap}{%

```

```

{name}{name},%
{sort}{sortvalue},% unescaped sort value
{type}{type},%
{first}{first},%
{firstplural}{firstpl},%
{text}{text},%
{plural}{plural},%
{description}{desc},%
{descriptionplural}{descplural},%
{symbol}{symbol},%
{symbolplural}{symbolplural},%
{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
      {%
        \edef#1{\expandafter\@secondoftwo\@gls@map}%
        \@endfortrue
      }%
    }%
  }%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
      {\csname gls@assign@#1@field\endcsname}[2]{%
        \@gls@expand@field{##1}{#1}{##2}%
      }%
  }%
  {}%
  \@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%

```

```

{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{,}{#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
{}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{,}{#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @gls@user@#1\endcsname}[2][ ]{%
\new@ifnextchar[%
{\csuse{@gls@user@#1@}{##1}{##2}}%
}
}
}

```

```

        {\csuse{@gls@user@#1@}{##1}{##2}[]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
  '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2}[]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
  '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1\endcsname}[2] []{%
    \new@ifnextchar[%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%
}%
{%
  \PackageError{glossaries}{Key '#1' already exists}{}%
}%
}

```

```

\newcommand{\glsfieldxddef}[3]{%
\glsdoifexists{#1}%
{%
\edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\protected@csxdef{glo@\@glo@label @#2}{#3}%
}%
}%
\PackageError{glossaries}{Key '#2' doesn't exist}{}%
}%
}
\newcommand{\glsfielddedef}[3]{%
\glsdoifexists{#1}%
{%
\edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\protected@csedef{glo@\@glo@label @#2}{#3}%
}%
}%
\PackageError{glossaries}{Key '#2' doesn't exist}{}%
}%
}
\newcommand{\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{\glsfieldddef}[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{\ifglsfielddfeq}[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@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
}

```

```

\def\@glo@parent{}%
\def\@glo@prefix{}%
\@gls@initnonumberlist
\def\@glo@useri{}%
\def\@glo@userii{}%
\def\@glo@useriii{}%
\def\@glo@useriv{}%
\def\@glo@userv{}%
\def\@glo@uservi{}%
\def\@glo@short{}%
\def\@glo@shortpl{}%
\def\@glo@long{}%
\def\@glo@longpl{}%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glolist@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\@empty
  \fi
}%
}%
\protected@edef\@glolist@{\csname glolist@\@glo@type\endcsname}%
\expandafter\xdef\csname glolist@\@glo@type\endcsname{%
  \@glolist@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%

```

```

{
  \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
  \def\@glo@parent{}%
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
{
  \ifglentryexists{\@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\@glsdefault@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\@glsdefault@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else

```

```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @first\endcsname\glspluralsuffix}%
  {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glo@type@\glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glo@type@\glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
  \@glsnoname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\glo@label @numberlist}%
{%
  \csxdef{glo@\glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \@glsnodesc
  \let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter

```

```

\gls@assign@field\expandafter
{\csname glo@\@glo@label @symbol\endcsname}
{\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\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}%
{%
\ifdefequal\@glo@thislabel\glo@label
}{\@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]{%
  \edef\@glo@esclabel{#1}%
  \@gls@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gls@checkmkidxchars\@glo@sort
  \@gls@checkmkidxchars\@glo@prefix
  \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \ifglxindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string\@glo@sort\string" %
          \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
        }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
          \string"\@glo@prefix\@glossarysubentryfield
            {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
        }%
    \fi
  \else
    \ifx\@glo@parent\@empty
      \@onelevel@sanitize\@glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossaryentryfield{\@glo@esclabel}%
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossarysubentryfield
          {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
      }%
    \fi
  \fi
\fi

```

```

}
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}%
  {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
  {}}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
  \ifmeasuring@
  \else
  #1%
  \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\glspatchtabularx\relax
  }%
  {}}%
}
\newcommand*{\glsreset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsreset{#1}%
    }%
  }%
}
\newcommand*{\glslocalreset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalreset{#1}%
    }%
  }%
}
\newcommand*{\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}

```

```

}
\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
}

```

```

\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*{\glsenableentrycount}{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
  \newcommand*{\glsentrycurrcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
    {0}{\@gls@entry@field{##1}{currcount}}%
  }%
  \newcommand*{\glsentryprevcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
    {0}{\@gls@entry@field{##1}{prevcount}}%
  }%
  \renewcommand*{\@glsunset}[1]{%
    \@glsunset{##1}%
    \@gls@increment@currcount{##1}%
  }%
  \renewcommand*{\@glslocalunset}[1]{%
    \@glslocalunset{##1}%
    \@gls@local@increment@currcount{##1}%
  }%
  \renewcommand*{\@glsreset}[1]{%
    \@glsreset{##1}%
    \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
  }%
  \renewcommand*{\@glslocalreset}[1]{%
    \@glslocalreset{##1}%
    \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
  }%
  \def\@cgls@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
      \cglsformat{##2}{##3}%
      \glsunset{##2}%
    \else
      \@gls@{##1}{##2}[##3]%
    \fi
  }%
  \def\@cglspl@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
      \cglsplformat{##2}{##3}%

```

```

        \glsunset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
\def\@cGls@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cGlsformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@cGlspl@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax
        \cGlsplformat{##2}{##3}%
        \glsunset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@glsdetoklabel{##1}@currcount}{%
        \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*\string\@gls@entry@count}[2]{}}%
    \forallglsentries{\@glsentry}{%
        \ifglsused{\@glsentry}%
        {\immediate\write\@auxout
            {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
        {}%
    }%
}
\newcommand*{\@gls@entry@count}[2]{}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][ ]{%
    \new@ifnextchar[{\@cGls@{##1}{##2}}{\@cGls@{##1}{##2}[]}%
}

```

```

}
\def\@cgl#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
    \string\gl\space since you haven't enabled entry counting}%
  \@gl#1}{#2} [#3]%
}
\newcommand*\@cglformat}[2]{%
  \ifglshaslong{#1}{\glentrylong{#1}}{\glentryfirst{#1}}#2%
}
\newrobustcmd*\@cGl{\@glshypopt\@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}%
  \@Gl#1}{#2} [#3]%
}
\newcommand*\@cGlformat}[2]{%
  \ifglshaslong{#1}{\Glentrylong{#1}}{\Glentryfirst{#1}}#2%
}
\newrobustcmd*\@cGlsp{\@glshypopt\@cGlsp}
\newcommand*\@cGlsp}[2] []{%
  \new@ifnextchar[{\@cGlsp#1}{#2}]{\@cGlsp#1}{#2} []}%
}
\def\@cGlsp#1#2[#3]{%
  \GlossariesWarning{\string\cGlsp\space is defaulting to
    \string\glsp\space since you haven't enabled entry counting}%
  \@glsp#1}{#2} [#3]%
}
\newcommand*\@cGlspformat}[2]{%
  \ifglshaslong{#1}{\glspentrylongpl{#1}}{\glspentryfirstplural{#1}}#2%
}
\newrobustcmd*\@cGlsp{\@glshypopt\@cGlsp}
\newcommand*\@cGlsp}[2] []{%
  \new@ifnextchar[{\@cGlsp#1}{#2}]{\@cGlsp#1}{#2} []}%
}
\def\@cGlsp#1#2[#3]{%
  \GlossariesWarning{\string\cGlsp\space is defaulting to
    \string\Glsp\space since you haven't enabled entry counting}%
  \@Glsp#1}{#2} [#3]%
}
\newcommand*\@cGlspformat}[2]{%
  \ifglshaslong{#1}{\Glspentrylongpl{#1}}{\Glspentryfirstplural{#1}}#2%
}
\newcommand*\loadglentries}[2] [\@gl@default]{%
  \let\@gl@default\gldefaulttype
  \def\gldefaulttype{#1}\input{#2}%
  \let\gldefaulttype\@gl@default
}

```

```

\onlypreamble{\loadglsentries}
\newcommand*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{%
  \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
  }%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
  }%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirstplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    \xmakefirstuc\@glo@etext
  }%
}

```

```

    {%
      #1{\Glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
    {%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
    }%
  }%
}

```

```

        {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
}%
{%
    \ifbool{glscompatible-3.07}%
    {%
        \protected@edef\@glo@etext{%
            #1{\glsentryfirst{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}}%
        \xmakefirstuc\@glo@etext
    }%
    {%
        #1{\Glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
}%
}%
}%
{%
    \ifglsused\glslabel
    {%
        \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
    {%
        \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
}%
}%
}%
}%
{%
    \ifglsused{\glslabel}%
    {%
        #2{\glscustomtext}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{%
    }%
    {%
        #1{\glscustomtext}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{%
    }%
    }%
}%
}
\newcommand*{\glsgenentryfmt}{%
    \ifdefempty\glscustomtext

```

```

{%
  \glsifplural
  {%
    \glscapscase
    {%
      \ifglsused\glslabel
      {%
        \glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \Glsentryplural{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirstplural{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \mfirstucMakeUppercase
      {\glsentryplural{\glslabel}\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\glsentryfirstplural{\glslabel}\glsinsert}%
    }%
  }%
}
}
}
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glsentrytext{\glslabel}\glsinsert
    }%
    {%
      \glsentryfirst{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \Glsentrytext{\glslabel}\glsinsert
    }%
  }%
}

```



```

\makefirstuc\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
      \ifglssacrfootnote
        \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@glssisablehyperinlist{%
  \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@glssisablehyperinlist
  \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\@nil}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@nil
\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@checkesbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%

```

```

\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2\""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\#"#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?#\?#\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null

```

```

\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeschar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
\def\@gls@checkeschar{\@gls@checkeschar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkeschar{\@gls@checkeschar#2\|\|\null}%
\else
\def\@gls@checkeschar{\@gls@checkeschar#2\|#3\null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkeslevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkeslevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkeslevel{\@gls@checkeslevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkeslevel{\@gls@checkeslevel#2\!\!\null}%
\else
\def\@gls@checkeslevel{\@gls@checkeslevel#2\|#3\null}%
\fi
\fi
\@gls@checkeslevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!|\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@gls@checkactual{\@gls@checkactual#3\null}%

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\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}%

```

```

\newcommand{\gls@xdycheckbackslash}{%
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
  \noexpand\fi
\noexpand\else
\newcommand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\newcommand\def\noexpand\@gls@xdycheckbackslash{%
  \noexpand\@gls@xdycheckbackslash##2\@backslashchar
  \@backslashchar\noexpand\null}%
\noexpand\else
\newcommand\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\gls caps case\@firstofthree
\let\gls custom text\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
\glslocalunset{#2}%
\else
\glsunset{#2}%
\fi
}%
\gls post link hook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][{}]{%
\new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}]%
}
\def\@Gls@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls caps case\@secondofthree
\let\gls custom text\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
\glslocalunset{#2}%
\else
\glsunset{#2}%
\fi
}

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\newcommand*{\@GLS}[2][]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls@scaps@case\@thirdofthree
    \let\gls@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\gls@scaps@case\@firstofthree
    \let\gls@customtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \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\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\@GLspl}{\@gl@hyp@opt\@GLspl}
\newcommand*{\@GLspl}[2][{}]{%
  \new@ifnextchar[{\@GLspl@{#1}{#2}}{\@GLspl@{#1}{#2}[]]}%
}
\def\@GLspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\@glsdisp}{\@gl@hyp@opt\@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{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  }%
}

```

```

    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@gls@link@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*\@gls@hyp@opt\@gls@hyp@opt
\newcommand*\@gls@hyp@opt[2][]{%
  \new@ifnextchar[{\@gls@hyp@opt@#1}{#2}]{\@gls@hyp@opt@#1}{#2}[]}}
\def\@gls@hyp@opt@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*\@gls@GL@opt\@gls@GL@opt
\newcommand*\@gls@GL@opt[2][]{%
  \new@ifnextchar[{\@gls@GL@opt@#1}{#2}]{\@gls@GL@opt@#1}{#2}[]}}
\def\@gls@GL@opt@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*\@gls@G@opt\@gls@G@opt
\newcommand*\@gls@G@opt[2][]{%
  \new@ifnextchar[{\@gls@G@opt@#1}{#2}]{\@gls@G@opt@#1}{#2}[]}}
\def\@gls@G@opt@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*\@gls@f@opt\@gls@f@opt
\newcommand*\@gls@f@opt[2][]{%
  \new@ifnextchar[{\@gls@f@opt@#1}{#2}]{\@gls@f@opt@#1}{#2}[]}}
\def\@gls@f@opt@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*\@gls@F@opt\@gls@F@opt
\newcommand*\@gls@F@opt[2][]{%
  \new@ifnextchar[{\@gls@F@opt@#1}{#2}]{\@gls@F@opt@#1}{#2}[]}}
\def\@gls@F@opt@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*\@gls@GLS@opt\@gls@GLS@opt
\newcommand*\@gls@GLS@opt[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*{\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}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] []{%
  \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] []{%
  \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}]
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] []{%
  \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] []{%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] []{%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2] []{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2} []}]
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2] []{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] []{%

```

```

    \new@ifnextchar [ {\@glsuserii@{#1}{#2}} {\@glsuserii@{#1}{#2} [] }
\def \@glsuserii@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuserii{#2}#3}%
}
\newrobustcmd* {\Glsuserii} {\@gls@hyp@opt \@Glsuserii}
\newcommand* {\@Glsuserii}[2] [] {%
    \new@ifnextchar [ {\@Glsuserii@{#1}{#2}} {\@Glsuserii@{#1}{#2} [] }
\def \@Glsuserii@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\Glsentryuserii{#2}#3}%
}
\newrobustcmd* {\GLSuserii} {\@gls@hyp@opt \@GLSuserii}
\newcommand* {\@GLSuserii}[2] [] {%
    \new@ifnextchar [ {\@GLSuserii@{#1}{#2}} {\@GLSuserii@{#1}{#2} [] }
\def \@GLSuserii@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd* {\glsuseriii} {\@gls@hyp@opt \@glsuseriii}
\newcommand* {\@glsuseriii}[2] [] {%
    \new@ifnextchar [ {\@glsuseriii@{#1}{#2}} {\@glsuseriii@{#1}{#2} [] }
\def \@glsuseriii@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\glsentryuseriii{#2}#3}%
}
\newrobustcmd* {\GLSuseriii} {\@gls@hyp@opt \@GLSuseriii}
\newcommand* {\@GLSuseriii}[2] [] {%
    \new@ifnextchar [ {\@GLSuseriii@{#1}{#2}} {\@GLSuseriii@{#1}{#2} [] }
\def \@GLSuseriii@#1#2[#3] {%
    \@gls@field@link{#1}{#2} {\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}{\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}{\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}{\mfirstucMakeUppercase{\glsentryuseriv{#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}{\glsentryuserivi{#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}{\Glsentryuserivi{#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{\glsentryuserivi{#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{\glentryshort{#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\gls caps case\@secondofthree
        \let\glsinsert\@empty
        \def\gls custom text{%
            \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\gls caps case\@thirdofthree
        \let\glsinsert\@empty
        \def\gls custom text{%
            \mfirstucMakeUppercase{\acronymfont{\glentryshort{#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\gl@ifplural\@firstoftwo
  \let\glscapscase\@firstofthree
  \let\gl@insert\@empty
  \def\glscustomtext{%
    \acronymfont{\gl@entryshortpl{#2}}#3%
  }%
  \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gl@hyp@opt\ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] []{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\gl@insert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gl@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\gl@insert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\gl@entryshortpl{#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\glsescapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2][{}]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\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\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acr longpl}{\@gls@hyp@opt\ns@Acr longpl}
\newcommand*{\ns@Acr longpl}[2][]{%
  \new@ifnextchar[{\@Acr longpl{#1}{#2}}{\@Acr longpl{#1}{#2} []}%
}
\def\@Acr longpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACR longpl}{\@gls@hyp@opt\ns@ACR longpl}
\newcommand*{\ns@ACR longpl}[2][]{%
  \new@ifnextchar[{\@ACR longpl{#1}{#2}}{\@ACR longpl{#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@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field '#2' doesn't exist for glossary
        entry '\glsdetoklabel{#1}'}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  ??%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acrentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
  }%
}

```

```

\expandafter\@gls@getbody\@glo@text{}\@nil
\expandafter\ifx\@gls@body\glsentrylong\relax
\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newrobustcmd*\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*\glsentrydescplural}[1]{%
\@gls@entry@field{#1}{descplural}%
}
\newrobustcmd*\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newrobustcmd*\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*\glsentryplural}[1]{%
\@gls@entry@field{#1}{plural}%
}
\newrobustcmd*\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*\glsentrysymbol}[1]{%

```

```

    \@gls@entry@field{#1}{symbol}%
  }
  \newrobustcmd*{\Glsentrysymbol}[1]{%
    \@Gls@entry@field{#1}{symbol}%
  }
  \newcommand*{\glsentrysymbolplural}[1]{%
    \@gls@entry@field{#1}{symbolplural}%
  }
  \newrobustcmd*{\Glsentrysymbolplural}[1]{%
    \@Gls@entry@field{#1}{symbolplural}%
  }
  \newcommand*{\glsentryfirst}[1]{%
    \@gls@entry@field{#1}{first}%
  }
  \newrobustcmd*{\Glsentryfirst}[1]{%
    \@Gls@entry@field{#1}{first}%
  }
  \newcommand*{\glsentryfirstplural}[1]{%
    \@gls@entry@field{#1}{firstpl}%
  }
  \newrobustcmd*{\Glsentryfirstplural}[1]{%
    \@Gls@entry@field{#1}{firstpl}%
  }
  \newrobustcmd*{\@glsentrytitlecase}[2]{%
    \glsdoifexists{#1}%
    {%
      \glsfieldfetch{#1}{#2}{\@gls@value}%
      \xcapitalisewords{\@gls@value}%
    }%
  }
  }
  \ifdef\texorpdfstring
  {
    \newcommand*{\glsentrytitlecase}[2]{%
      \texorpdfstring
        {\@glsentrytitlecase{#1}{#2}}%
        {\@gls@entry@field{#1}{#2}}%
    }
  }
  {
    \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
  }
  \newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
  \newcommand*{\glsentrysort}[1]{%
    \@gls@entry@field{#1}{sort}%
  }
  \newcommand*{\glsentryparent}[1]{%
    \@gls@entry@field{#1}{parent}%
  }
  \newcommand*{\glsentryuseri}[1]{%
    \@gls@entry@field{#1}{useri}%
  }

```

```

}
\newrobustcmd*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*{\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*{\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*{\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*{\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*{\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*{\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*{\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*{\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*{\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*{\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*{\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*{\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*{\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
}

```

```

\newcommand*\glsentryfull}[1]{%
  \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*\Glsentryfull}[1]{%
  \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*\Glsentryfullpl}[1]{%
  \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
\@ifpackageloaded{hyperref} {%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup
      \edef\@glo@label{\glsdetoklabel{#1}}%
      \let\@org@glsnumberformat\glsnumberformat
      \def\glsnumberformat##1{##1}%
      \protected@edef\the@numberlist{%
        \csname glo@\@glo@label @numberlist\endcsname}%
      \def\@gls@numlist@sep{}%
      \def\@gls@numlist@nextsep{}%
      \def\@gls@numlist@lastsep{}%
      \def\@gls@thislist{}%
      \def\@gls@donext@def{}%
      \renewcommand\do[1]{%
        \protected@edef\@gls@thislist{%
          \@gls@thislist
          \noexpand\@gls@numlist@sep
          ##1%
        }%
      }%
    }%
  }%
}

```

```

        \let\@gls@numlist@sep\@gls@numlist@nextsep
        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
            \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
    }%
}
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
    \def\@glo@label{#2}%
    \@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][ ]{%
    \@gls@adjustmode
    \glsdoifexists{#2}%
    {%
        \def\@glsnumberformat{glsnumberformat}%
        \edef\@gls@counter{\csname glo@%glsdetoklabel{#2}@counter\endcsname}%
        \setkeys{glossadd}{#1}%
        \@gls@saveentrycounter
        \@gls@setsort{#2}%
        \@do@wrglossary{#2}%
    }%
}
}
\newcommand*{\@gls@adjustmode}{-}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][ ]{%
    \edef\@glo@type{\@glo@types}%
    \setkeys{glossadd}{#1}%
    \forallglsentries[\@glo@type]{\@glo@entry}{%
        \glsadd[#1]{\@glo@entry}%
    }%
}
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glo@entry}%
    {%
        \ifglsused{\@glo@entry}{-}{\glsadd[format=glsignore]{\@glo@entry}}%
    }%
}
}
\newcommand*{\glsignore}[1]{-}

```

```

\edef\glsopenbrace{\expandafter@gobble\string\{ }
\edef\glsclosebrace{\expandafter@gobble\string\}}
\edef\glsbackslash{\expandafter@gobble\string\}
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\%}
\edef\glsstildechar{\string~}
\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]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else

```

```

\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
  \glsnoxindywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \@gls@addpredefinedattributes
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
    \protected@write\glswrite{}{(require
      \string"\@xdystyle.xdy\string")}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{}{(define-location-class
      \string"\@gls@classI\string"^^J\space\space\space
      (
        :sep "{}"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
    )
  }%
  {%
  \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{}{(define-location-class
      \string"\@gls@classII-\@gls@classI\string"
      ^^J\space\space\space
      (
        :sep "{"
        \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
        :sep "}"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
    }%
  }%
}

```

```

    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected\write\glswrite{}{\string\providecommand*
      \expandafter\string
      \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
      {%
        \string\setentrycounter
        [\expandafter@gobble\string\#1]{\@this@ctr}%
        \expandafter\string
        \csname\@this@attr\endcsname
        {\expandafter@gobble\string\#2}%
      }%
    }%
  }%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glspercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)%}

```

```

\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote

```

```

\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string}
\write\glswrite{item_0 \string\string%\string\n\string}
\write\glswrite{item_1 \string\string%\string\n\string}
\write\glswrite{item_2 \string\string%\string\n\string}
\write\glswrite{item_01 \string\string%\string\n\string}
\write\glswrite{item_x1
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glsresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\{\string}}
\write\glswrite{heading_suffix
  \string\string}\string\relax
  \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string\glsymbols\string}
\write\glswrite{numhead_positive \string\glsnumbers\string}
\write\glswrite{page_compositor \string\glscompositor\string}

```

```

\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremake\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
  \IfTrackedLanguage{german}%
  {%
    \def\@gls@extramakeindexopts{-g}%
  }%
  {}%
}%
{}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else

```

```

\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \noexpand\@gls@quotearch\noexpand\@gls@quotearch}%
\noexpand\ifx\noexpand\null####3\noexpand\null
  \noexpand\def\noexpand\@gls@checkquote{%
    \noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
  \noexpand\def\noexpand\@gls@checkquote{%
    \noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
  \noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
  #1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesquote####1\noexpand\@nil
  \expandonce{\cename#1\endcename}\expandonce{\cename#1\endcename}%
  \noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesactual####1\noexpand\@nil
  \noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
  \noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
  \noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesubar####1\noexpand\@nil
  \noexpand\\|noexpand\\|noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil

```

```

        \noexpand!\noexpand!\noexpand\null
        \noexpand\expandafter\noexpand\@gls@updatechecked
        \noexpand\@gls@checkedmkidx{###1}%
    }%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
    \noexpand\def\noexpand\@gls@checkescquote###1%
        \expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
        ###3\noexpand\null{%
            \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
            \noexpand\toks@={###1}%
            \noexpand\ifx\noexpand\null###2\noexpand\null
            \noexpand\ifx\noexpand\null###3\noexpand\null
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
                \noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
                \noexpand\else
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
                    \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                        \csname#1\endcsname}\noexpand\@gls@quotechar
                    \noexpand\@gls@quotechar\noexpand\string\expandonce{%
                        \csname#1\endcsname}\noexpand\@gls@quotechar}%
                \noexpand\def\noexpand\@gls@checkescquote{%
                    \noexpand\@gls@checkescquote###3\noexpand\null}%
                \noexpand\fi
            \noexpand\else
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
                    \noexpand\@gls@quotechar\noexpand\string
                    \expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
                \noexpand\ifx\noexpand\null###3\noexpand\null
                \noexpand\def\noexpand\@gls@checkescquote{%
                    \noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
                    \expandonce{\csname#1\endcsname}\noexpand\null}%
                \noexpand\else
                \noexpand\def\noexpand\@gls@checkescquote{%
                    \noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
                    ###3\noexpand\null}%
                \noexpand\fi
            \noexpand\fi
        \noexpand\@gls@checkescquote
    }%
}%
\@gls@docheckquotedef
}
\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as

```

```

        soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*{\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand*{\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%
    \PackageError{glossaries}%
    {Glossary type ‘#1’ not defined}%
    {New glossaries must be defined before using \string\makeglossaries}%
  }%
}
\newcommand*{\@glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
    \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
\ifglxindy
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\@empty}%
    \ifcsundef{@xdy@#1@language}%
    {\let\@gls@langmod\@xdy@main@language}%
    {\letcs\@gls@langmod{@xdy@#1@language}}%
    \edef\@gls@dothiswrite{\noexpand\write18{xindy
      -I xindy
      \@gls@order
      -L \@gls@langmod\space
      -M \gls@istfilebase\space
      -C \gls@codepage\space

```

```

        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}
{
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}
}
\else
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-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
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \ifcsundef{@xdy@#1@language}%
            {\let\@gls@langmod\@xdy@main@language}%
            {\letcs\@gls@langmod{@xdy@#1@language}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space
                -C \gls@codepage\space
                -t \jobname.\csuse{@glotype@#1@log}
            }
        }
    }
}

```

```

        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{@@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-1 }}%
            {\let\@gls@order\@empty}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
            -s \istfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}%
            }%
            \@gls@dothiswrite
        }%
        {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
        doesn't exist. Rerun may be required}}%
    }%
    {%
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\fi
\newcommand*{@@warn@nomakeglossaries}{}
\newcommand*{@@warn@nomakeglossaries}{@@warn@nomakeglossaries}
\newcommand{@@gls@automake@immediate}{%
    \ifnum\gls@automake@nr=2\relax
    \@for\@gls@type:=\@gls@types\do{%
        \ifdefempty{\@gls@type}{%
            {\@gls@automake@immediate{\@gls@type}}%
        }%
    }
    \glsautomakefalse
    \renewcommand*{@@gls@doautomake}{}%
}
\fi
}
\newcommand*{\makeglossaries}{%

```

```

\domakeglossaries
{%
  \gls@automake@immediate
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \protected@write\@auxout{}{\string\providecommand\string\glsorder[1]{} }
  \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
  \ifundef\@gls@extramakeindexopts
  {}%
  {%
    \protected@write\@auxout{}{\string\providecommand
      \string\gls@extramakeindexopts[1]{} }
    \protected@write\@auxout{}{\string\gls@extramakeindexopts
      {\@gls@extramakeindexopts}}%
  }%
  \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
  \protected@write\@auxout{}{\string\glsorder{\glsorder}}
  \@for\@glo@type:=\@glo@types\do{%
    \ifthenelse{\equal{\@glo@type}{}}{}{}%
    \makeglossary{\@glo@type}%
  }%
  \renewcommand*\newglossary[4] []{%
  \PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
  \let\@makeglossary\@gobble
  \let\makeglossaries\relax
  \@disable@onlypremakeg
  \let\gls@checkseeallowed\relax
  \let\warn@nomakeglossaries\relax
  \def\warn@noprintglossary{%
    \ifdefstring{\@glo@types}{,}%
    {%
      \GlossariesWarningNoLine{No glossaries have been defined}%
    }%
    {%
      \GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
    }%
  }%
  \ifglssavenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
  {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
  \fi
  \let\makenoidxglossaries\@no@makeglossaries
  \renewcommand*\@printgloss@setsort{%

```

```

\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
\@domakeglossaries
{%
\renewcommand{\@gls@nofwarn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
\let\@gls@checkmkidxchars\@gobble
\glsesclocationsfalse
\let\@do@wrglossary\gls@noidxglossary
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\@gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
\edef\@gls@label{\glsdetoklabel{##1}}%
\protected@write\@auxout{}{%
\string\@gls@reference
{\csname glo@\@gls@label @type\endcsname}%
{\@gls@label}%
}%
\string\glsseeformat##2}%
}%
}%
\AtBeginDocument
{%
\write\@auxout{\string\providecommand\string\@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
\GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J

```

```

    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
  defined in the preamble^^Jwhen you use
  \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
  \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
      a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%

```

```

    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc@glsnoidxdisplayloc
  \let\@gls@org@glsseeformat@glsseeformat
  \let@glsnoidxdisplayloc##2\relax
  \let@glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let@glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let@glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}

```

```

\newcommand*\gls@noidxglossary}{%
  \protected@write\@auxout}{%
    \string\gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@glslocref}%
    }%
  }%
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*\@glswritefiles{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%
        \expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
      \immediate\closeout\glswrite
    \fi
  }%
}
\newcommand*\gls@glossary}[1]{%
  \@gls@glossary{#1}%
}
\newcommand*\@gls@glossary}[2]{%

```

```

\if@gls@debug
  \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
\fi
}
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begingroup\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
      {%
        \expandafter\protected@write\csname glo@#1@file\endcsname{%
          \gls@disablepagerefexpansion}{#2}%
        }%
      {%
        \ifignoredglossary{#1}{}%
          {%
            \GlossariesWarning{No file defined for glossary ‘#1’}%
          }%
        }%
      }%
    \fi
  \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
  \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*{\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
}
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}

```

```

\newcommand*\gls@romanpage{\romannumeral\c@page}
\newcommand*\gls@Romanpage{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
  \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
  \eappto\@wrglossarynumberhook{%
    \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
    \expandonce{\csname#1\endcsname}%
    \noexpand\def\expandonce{\csname#1\endcsname}{%
      \noexpand\@wrglossary@pageformat
      \expandonce{\csname gls#1page\endcsname}%
      \expandonce{\csname org@gls#1\endcsname}%
    }%
  }%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*\@@do@wrglossary}[1]{%
  \ifglsclocations
    \@@do@esc@wrglossary{#1}%
  \else
    \@@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*\@@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@gls@locref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@gls@Hlocref\expandafter{\theHglentrycounter}%
  \ifx\@gls@Hlocref\@gls@locref
    \def\@gls@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@gls@locref}{\@gls@Hlocref}}%
  }%
  \@do@gls@getcounterprefix
\fi
\edef\@gls@label{\gls@detoklabel{#1}}%
\@@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*\@@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph

```

```

\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
  \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
  \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
  \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theHglentrycounter\theglsentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@glsHlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@do@@wrglossary
}
\newcommand*{\@do@@wrglossary}{%
\ifglsxindy
\expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
\def\@glo@range{}%
\expandafter\if\@glo@prefix(\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

```

```

        :locref \string"{\@glo@counterprefix}{\@glslocref}\string" %
        :attr \string"@gls@counter@glo@suffix\string"
        \@glo@range
    )
    }%
\else
    \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
    {\@glo@counterprefix}%
    \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    \string\glossaryentry{\csname glo@\@gls@label @index\endcsname
    \@gls@encapchar@glo@numfmt}{\@glslocref}}%
\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
    \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
    \ifx\@gls@thisloc\@gls@thisHloc
        \def\@glo@counterprefix{%
    \else
        \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
            \def\@glo@tmp{##2}%
            \ifx\@glo@tmp\@empty
                \def\@glo@counterprefix{%
            \else
                \def\@glo@counterprefix{##1}%
            \fi
        }%
    \@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
    \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
    prefixing^^Jlocation ‘#1’. You need to modify the
    definition of \string\theH@gls@counter^^Jotherwise you
    will get the warning: "‘name{\@gls@counter.#1}’ has been^^J
    referenced but does not exist"%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
    \gls@glossary{\csname glo@#1@type\endcsname}{%
        (indexentry
            :tkey (\csname glo@#1@index\endcsname)
            :xref (\string"@gls@xref\string")
            :attr \string"see\string"
        )
    }%
\else
    \gls@glossary{\csname glo@#1@type\endcsname}{%

```

```

\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*\@glssee}[3][\seename]{%
\do@seeglossary{#2}{[#1]{#3}}}
\newcommand*\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*\@glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}
\newrobustcmd*\@glsseelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext
\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseelastsep
\let\@gls@donext\glsseesep
}%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep}{, }
\newrobustcmd*\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*\gls@save@numberlist}[1]{%
\ifglssavenumberlist
\toks@{#1}%
\edef\@do@writeaux@info{%
\noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\@do@writeaux@info
\protected@write\@auxout{}{\@do@writeaux@info}%
\fi
}
\newcommand*\warn@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@dotocitle{\glssettocitle{\@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@dotocitle
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\egroup
\global\let\glossaryentrynumbers\org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglsxindy
    \ifcsundef{xdy@\@glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdy@language[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%

```

```

        \string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
    }%
}%
}%
{%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
@language\endcsname}}%
}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@gls@codepage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
}%
}%
\@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
\GlossariesWarningNoLine{\string\makeglossaries\space
hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
\glosortentrieswarning
\def\@glo@sortinglist{}%
\def\@glo@sortinghandler{#1}%
\edef\@glo@type{#2}%
\forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
\csdef{\@glsref@#2}{%
\for\@this@label:=\@glo@sortinglist\do{%
\xifinlistcs{\@this@label}{\@glsref@#2}%
}%
{%
\listcsxadd{\@glsref@#2}{\@this@label}%
}%
\ifcsdef{\@glo@sortingchildren@\@this@label}%
{%
\@glo@addchildren{#2}{\@this@label}%
}%
}%
}%
}%

```

```

}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
    {}%
    {%
      \listcsxadd{\@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{\@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    {}%
  }%
}
\egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \edef\@glo@parent{\csuse{glo\@glsdetoklabel{#1}@parent}}%
    \ifcsundef{\@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{\@glo@sortingchildren@\@glo@parent}{}%
    }%
    {}%
    \expandafter\@glo@sortedinsert
    \csname \@glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{\@glsref@\@glo@type}%
    {}%
    {}%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
  {}%
  \@glo@sortedinsert{\@glo@sortinglist}{#1}%
}
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo\@glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo\@glsdetoklabel{#2}@sort}%
  \edef\@glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
  }%
}

```

```

        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@word}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=word]}{ }%
    }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
    }%
}

```

```

}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=letter]}{}%
}%
}
\newcommand*\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@\glsorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=standard]}{}%
  }%
}
\newcommand*\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=case]}{}%
  }%
}
\newcommand*\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
  }%
}
\newcommand*\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{%
  \for\glsentries[#1]{\@gls@thislabel}%
  }%

```

```

\ifinlistcs{\@gls@thislabel}{\@glsref@#1}%
{%
  \listeadadd{\@glo@sortinglist}{\@gls@thislabel}%
}%
{%
}%
}%
\cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{%
        }%
      \glossarysection[\glossarytoctitle]{\glossarytitle}%
      \glossarypreamble
      \def\@gls@currentlettergroup{}%
      \begin{theglossary}%
      \glossaryheader
      \glsresetentrylist
      \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
      \end{theglossary}%
      \glossarypostamble
    }%
    {%
      \@gls@noref@warn{\@glo@type}%
    }%
  }
}
\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%

```

```

\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\uccode‘#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@=#1\relax
}%
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{\glo@\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{\@gls@loclist}
    }%
  }%
}
}
\newcommand*\@gls@noidxloclist[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\@gls@noidxloclisthandler}{#1}%
}
\newcommand*\@gls@noidxloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\@gls@noidxdisplayloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\@gls@noidxdisplayloc[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@glstetoklabel{#2}@loclist}%
  {\csgdef{glo@glstetoklabel{#2}@loclist}{}}%
  {}%
  \listcsgadd{glo@glstetoklabel{#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{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}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}
\newcommand*\glsentryitem}[1]{%
  \ifglentrycounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glsstepsubentry{#1}\glssubentrycounterlabel

```

```

\fi
}
\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{}{}%
}%
{%
\@gls@warnontheglossdefined
\renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
\toks@{#2}%
\protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
\{noexpand\glsnamefont
\{expandafter\expandonce\csname glo@#1@name\endcsname\}}%
\{expandafter\expandonce\csname glo@#1@desc\endcsname\}}%
\{expandafter\expandonce\csname glo@#1@symbol\endcsname\}}%
\the\toks@}%
}%
\@do@glossentry
}
\newcommand*\glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%
}%
}
\newcommand*\Glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsnamefont{\Glsentryname{#1}}%
}%
}
\newcommand*\glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\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}{\glsymbolsgroupname}}{\glssymbols}{%
\ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
\def\@glo@counterprefix{#1}%
\ifx\@glo@counterprefix\@empty
\def\@glo@counterprefix{.}%
\else
\def\@glo@counterprefix{.#1.}%
\fi
\def\glsentrycounter{#2}%
}
}
\newcommand*{\setglossarystyle}[1]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
}%
{%
\csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
\protected@edef\@glossary@default@style{#1}%
\fi
}
}
\newcommand*{\glossarystyle}[1]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
}%
{%

```

```

\GlossariesWarning
{Deprecated command \string\glossarystyle.^~J
 I recommend you switch to \string\setglossarystyle\space unless
 you want to maintain backward compatibility}%
\setglossentrycompatibility
\csname @glsstyle@#1\endcsname
\ifcsdef{@glscompstyle@#1}%
{\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
}%
}%
\ifx@glossary@default@style\relax
\protected@edef@glossary@default@style{#1}%
\fi
}
\newcommand{\newglossarystyle}[2]{%
\ifcsundef{@glsstyle@#1}%
{%
\expandafter\def\csname @glsstyle@#1\endcsname{#2}%
}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
}%
}
\newcommand{\renewglossarystyle}[2]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
}%
{%
\csdef{@glsstyle@#1}{#2}%
}%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
\def\glshypernumber#1{#1}%
}%
{%
\def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
\ifx\#1\%
\else
\@delimR#1\delimR\delimR\%
\fi
\ifx\#2\%
\else
#2%
\fi
\ifx\#3\%

```

```

\else
  \@glsnumber#3\@nil
\fi
}
\def\@delimR#1\delimR #2\delimR #3\{%
\ifx\#2\%
  \@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\x{\the\toks@}%
  \ifx\x\empty
  \else
    \hyperlink{\glsentrycounter\@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{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand*\oldacronym[4][\gls@label]{%

```

```

\def\gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
  \expandafter\edef\csname#1\endcsname{%
    \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
  }%
}%
\def\gls@label{#2}%
\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\glslongtok

```

```

\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand*\newacronym[4][{}]{%
    \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%

```

```

\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*\acronymsort}[2]{#1}
\newcommand*\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}
\newcommand*\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%

```

```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glsentrylong{##1}##2\space
  (\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \Glsentrylong{##1}##2\space
  (\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{##1}##2\space
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\space
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
  \ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

```

}
\newcommand*\glsacsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}

```

```

}
\newacronymstyle{sc-short-long}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sp-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%

```

```

    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%

```

```

\ifglshaslong{\glslabel}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \glsentrylongpl{\glslabel}\glsinsert
    }%
    {%
      \Glsentrylongpl{\glslabel}\glsinsert
    }%
    {%
      \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}%
{%
    \ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
    \glshyperfirstfalse
    \renewcommand*{\genacrfullformat}[2]{%
        \protect\firstacronymfont{\glsentryshort{##1}}##2%
        \protect\footnote{\glsentrylong{##1}}%
    }%
    \renewcommand*{\Genacrfullformat}[2]{%
        \firstacronymfont{\Glsentryshort{##1}}##2%
        \protect\footnote{\Glsentrylong{##1}}%
    }%
}

```

```

\renewcommand*\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}}}%
\renewcommand*\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%

```

```

}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\aclong

```

```

\let\aclp\acrlongpl
\let\Acl\Acrlong
\let\Aclp\Acrlongpl
\let\acf\acrfull
\let\acfp\acrfullpl
\let\Acf\Acrfull
\let\Acfp\Acrfullpl
\let\ac\gls
\let\Ac\Gls
\let\acp\glspl
\let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%
    \@gls@expand@field{##1}{descplural}{##2}%
  }%

```

```

}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\SetDefaultAcronymStyle{%
\@for\@gls@type:=\@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*\acrnohreffootnote}[3]{%
\footnote{#3}%
}
\newcommand*\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\gls@customtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\gls@genentryfmt}%
}%
{%
\firstacronymfont{\gls@genentryfmt}%
\ifgls@has@symbol{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
}%
\glsifplural
{\gls@entrysymbolplural{\glslabel}}%
}
}
}
}

```

```

        {\glsentrysymbol{\glslabel}}%
      }%
    }%
  }%
  {\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\the\glslongtok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
\renewcommand{\newacronym}[4][{}]{%
\ifx\glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%

```

```

\DeclareAcronymList{\@glo@type}%
\SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionFootnoteNewAcronymDef
}%
\@for\@gls@type:=\@gls@acronymlists\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]{\gls@genentryfmt}%
}
\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@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\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
  \ifgl@smallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \ifgl@smaller
      \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
  \fi
\renewcommand{\newacronym}[4][[]]{%
  \ifx\@gl@acronymlists\@empty
    \def\@gl@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@gl@type}%
    \SetDescriptionDUAAcronymDisplayStyle{\@gl@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook
  \DescriptionDUANewAcronymDef
}%
\for\@gl@type:=\@gl@acronymlists\do{%
  \SetDescriptionDUAAcronymDisplayStyle{\@gl@type}%
}%
}
\newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
  \def\gl@entryfmt[#1]{%
    \ifempty\gl@customtext
      {%
        \ifgl@used{\gl@label}%
      }%
    }%
  }%

```

```

\let\gls@org@insert\glsinsert
\let\glsinsert@empty
\acronymfont{\gls@genentryfmt}\gls@org@insert
}%
{%
\gls@genentryfmt
\ifgls@shassymbol{\gls@label}%
{%
\gls@ifplural
{%
\def\@glo@symbol{\gls@entrysymbolplural{\gls@label}}%
}%
{%
\def\@glo@symbol{\gls@entrysymbol{\gls@label}}%
}%
\space(\protect\firstacronymfont
{\gls@capscase
{\@glo@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
{}%
}%
}{\gls@customtext\glsinsert}%
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\gls@labeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acronymformat{\the\gls@shorttok}{\the\gls@longtok}},%
sort={\the\gls@shorttok},%
first={\the\gls@longtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\gls@shorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\gls@shorttok},%
shortplural={\the\gls@shorttok\noexpand\acrpluralsuffix},%
long={\the\gls@longtok},%
longplural={\the\gls@longtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\gls@keylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural

```

```

\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gls@acronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDescriptionAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \gslongtok{##4}%
    \newacronymhook
    \DescriptionNewAcronymDef
  }%
  \@for\@gls@type:=\@gls@acronymlists\do{%
    \SetDescriptionAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand{\acronymfont}[1]{\textsc{##1}}
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
    \fi
  \fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{%
    \ifdefempty\glscustomtext
      {%
        \let\gls@org@insert\glsinsert
        \let\glsinsert\@empty
        \ifglsused{\glslabel}%
          {%
            \acronymfont{\gls@genentryfmt}\gls@org@insert

```

```

}%
{%
  \firstacronymfont{\glsgenentryfmt}\gls@org@insert
  \ifglshaslong{\glslabel}%
  {%
    \expandafter\protect\expandafter\acrfootnote\expandafter
    {\@gls@link@opts}{\@gls@link@label}%
    {%
      \glsifplural
        {\glsentrylongpl{\glslabel}}%
        {\glsentrylong{\glslabel}}%
    }%
  }%
  }%
  {}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      first={\the\glsshorttok},%
      firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%
    \@gls@expand@field{##1}{descplural}{##2}%
  }%
}

```

```

}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\SetFootnoteAcronymStyle{%
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@gls@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@gls@type}%
\SetFootnoteAcronymDisplayStyle{\@gls@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 (##1)%
\fi
}%
}
\newcommand*\SetSmallAcronymDisplayStyle[1]{%
\defglsentryfmt[##1]{%

```

```

\ifdefempty\glscustomtext
{%
  \let\gls@org@insert\glsinsert
  \let\glsinsert@empty
  \ifglsused{\glslabel}%
  {%
    \acronymfont{\gls@genentryfmt}\gls@org@insert
  }%
  {%
    \gls@genentryfmt
    \ifgls@shassymbol{\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}}})%
    }%
    {}%
  }%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\SmallNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\glo@shortpl},%
    first={\the\glslongtok},%
    firstplural={\noexpand\expandonce\noexpand\glo@longpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\noexpand\glo@first},%
    descriptionplural={\noexpand\expandonce\noexpand\glo@longpl},%
    symbol={\the\glsshorttok},%
    symbolplural={\noexpand\expandonce\noexpand\glo@shortpl},%
  }%
}

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\@SetSmallAcronymStyle{%
\renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
        \SetSmallAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
    \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
    \renewcommand*\@acronymfont}[1]{\textsc{##1}}
    \renewcommand*\@acrpluralsuffix{\glsupacrpluralsuffix}%
\else
    \renewcommand*\@acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua

```

```

\ifglsacrsmallcaps
  \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
  \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
}%
\newcommand*{\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DUANewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry

```

```

\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDUADisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDUADisplayStyle{\@gls@type}%
}%
}
\newcommand*{\SetAcronymStyle}{%
\SetDefaultAcronymStyle
\ifglsacrdescription
\ifglsacrfootnote
\SetDescriptionFootnoteAcronymStyle
\else
\ifglsacrdua
\SetDescriptionDUAAcronymStyle
\else
\SetDescriptionAcronymStyle
\fi
\fi
\else
\ifglsacrfootnote
\SetFootnoteAcronymStyle
\else
\ifthenelse{\boolean{glsacrsmallcaps}\OR
\boolean{glsacrsmaller}}{%
\SetSmallAcronymStyle
}%
\fi
\ifglsacrdua
\SetDUASStyle
\fi
}%

```

```

    \fi
  \fi
}
\SetAcronymStyle
\newcommand*\SetCustomDisplayStyle[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*\CustomAcronymFields{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*\CustomNewAcronymDef{%
  \protected@edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \do@newglossaryentry
}
\newcommand*\SetCustomStyle{%
  \renewcommand*\newacronym[4][1]{%
    \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@glstetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showglotype}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@type\endcsname
}
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@counter\endcsname
}
\newcommand*\showglouseri}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useri\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriii\endcsname
}
\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@useriv\endcsname
}
}

```

```

\newcommand*\showglouerv}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouervi}[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 @glotype@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.46]
\fi

```

8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2020/03/19 v4.46 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*\PassOptionsToPackage{\CurrentOption}{glossaries}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx\@glsxtr@doaccsupp\empty
  \GlossariesWarning{The ‘glossaries-accsupp’
    package has been loaded\MessageBreak
    after the ‘glossaries-extra’ package. This\MessageBreak
    can cause a failure to integrate both packages. \MessageBreak
    Either use the ‘accsupp’ option when you load\MessageBreak
    ‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
    before loading ‘glossaries-extra’}%
  \fi
}
{}
\def\compatibleglossentry#1#2{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{%
    \noexpand\accsuppglossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@glossentry
}

```

```

}
\def\compatiblesubglossentry#1#2#3{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{%
    \noexpand\accsuppglossarysubentryfield{\number#1}%
    {#2}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@subglossentry
}
\RequirePackage{glossaries}[v4.46]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
  }
}
{}
\newcommand{\gls@accessibility}[4][[]]{%
  \@gls@showaccsupp{#1}{#2}{#3}%
  \gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
  \def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
  \def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%
  \def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
  \def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
  \def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
  \def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
  \def\@glo@symbolpluralaccess{#1}%
}
}

```

```

\define@key{glossentry}{descriptionaccess}{%
  \def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
  \def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
  \def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
  \def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
  \def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
  \def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
  \def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
  \def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
  \def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
  \def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
  \def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
  \def\@glo@userviaccess{#1}%
}
\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
}

```

```

{user1access}{useriaccess},%
{user2access}{useriiaccess},%
{user3access}{useriiiaccess},%
{user4access}{userivaccess},%
{user5access}{uservaccess},%
{user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
  \def\@glo@userivaccess{\relax}%
  \def\@glo@uservaccess{\relax}%
  \def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%
    \@glo@access}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
  \expandafter
  \protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
  \expandafter

```

```

\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
  \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
  \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
  \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
  \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
  \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
  \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
  \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
  \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
  \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @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*{\glentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*{\glentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*{\glentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*{\glentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*{\glentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*{\glentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*{\glentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*{\glentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*{\glentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*{\glentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*{\glentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%

```

```

\ifdef\glscategory
{%
  \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
  {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
  {%
    \ifcsdef{glsxtr\glscategory{#4}accsupp}%
    {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{gls#3accsupp}%
      {\csname gls#3accsupp\endcsname{#1}{#2}}%
      {\glsaccsupp{#1}{#2}}%
    }%
  }%
}%
{%
  \ifcsdef{gls#3accsupp}%
  {\csname gls#3accsupp\endcsname{#1}{#2}}%
  {\glsaccsupp{#1}{#2}}%
}%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@access}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}

```

```

\newrobustcmd*\glstextaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@textaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@pluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@firstaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@firstpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@symbolaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@symbolpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*\glsdescriptionaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@descaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@descpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@shortaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo\glsdetoklabel{#2}@shortpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}

```

```

\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
}
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
  }
}

```

```

\renewcommand*{\@@gls@default@entryfmt}[2]{%
\ifdefempty\glscustomtext
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}
{\glsinsert}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glsentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}%
}%
}%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\Glsentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{\glslabel}}{\glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}%
}%
}%
\ifglsused\glslabel

```

```

{%
  \MakeUppercase{%
    #2{\glspluralaccessdisplay
      {\glsentryplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glsymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
{%
  \MakeUppercase{%
    #1{\glsfirstpluralaccessdisplay
      {\glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glsymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glstextaccessdisplay
        {\glsentrytext{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glsymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
      }%
    {%
      #1{\glsfirstaccessdisplay
        {\glsentryfirst{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glsymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
      }%
    }%
  }%
  \ifglsused\glslabel
  {%
    #2{\glstextaccessdisplay
      {\Glsentrytext{\glslabel}}{\glslabel}}%

```

```

        {\glsdescriptionaccessdisplay
         {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glsymbolaccessdisplay
         {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}%
    }%
    {%
    #1{\glsfirstaccessdisplay
        {\Glsentryfirst{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
         {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glsymbolaccessdisplay
         {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}%
    }%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glsfirstaccessdisplay
    {\glsentrytext{\glslabel}}{\glslabel}}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
{%
\MakeUppercase{%
#1{\glsfirstaccessdisplay
    {\glsentryfirst{\glslabel}}{\glslabel}}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
}%
}%
{%
\ifglsused{\glslabel}%
{%
#2{\glsfirstaccessdisplay
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}
}
}

```

```

}%
{%
  #1{\glscustomtext}%
  {\glsdescriptionaccessdisplay
   {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glsymbolaccessdisplay
   {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}%
}%
}%
}
\renewcommand*{\glsentryfmt}{%
\ifdefempty\glscustomtext
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \ifglsused\glslabel
      {%
        \glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}%
        \glsinsert
      }%
      {%
        \glsfirstpluralaccessdisplay
        {\glsentryfirstplural{\glslabel}}{\glslabel}%
        \glsinsert
      }%
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}

```

```

}%
{%
  \glsfirstpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
    {\glslabel}%
  \mfirstucMakeUppercase{\glsinsert}%
}%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
    \glsinsert
  }%
  {%
    \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
    \glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glstextaccessdisplay
      {\mfirstucMakeUppercase{\glsentrytext{\glslabel}}{\glslabel}}%
      {\mfirstucMakeUppercase{\glsinsert}}%
  }%
  {%
    \glsfirstaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirst{\glslabel}}{\glslabel}}%
      {\mfirstucMakeUppercase{\glsinsert}}%
  }%
}%
}%
}%
{%
  \glscustomtext\glsinsert

```

```

    }%
  }
\renewcommand*{\glsngenacfmt}{%
  \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{\glshortentryshortpl{#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{\glshortaccessdisplay{\glshortentryshort{#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{\glshortaccessdisplay{\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{\glshortaccessdisplay
        {\MakeUppercase{\glshortentryshort{#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%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glsentrytext{#2}}{#2}}#3}%
}
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    \glstextaccessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}{#2}}%
    \mfirstucMakeUppercase{#3}}%
}
\def\@glsfirst@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glentryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glentryfirst{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glentryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glentryplural{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glentryfirstplural{#2}}{#2}#3}%
}
\def\@Glsfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glentryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glentryfirstplural{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{%
        {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glentryname{#2}}}{#2}%
        \mfirstucMakeUppercase{#3}}%
    }
}
\def\@glsdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}%
{\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\glsuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\glsuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

```

```

}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}
\def\@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}%
  {%
    \glsymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}
\newcommand*{\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glsymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*{\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glsymbolaccessdisplay{#5}{#2}}{#6}%
}

```

```

}
\renewacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
  \renewcommand*{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
  }%
}

```

```

\renewcommand*\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\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}%
}
\renewacronymstyle{long-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-short}%
}
{
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}{##1})}%
}
\renewacronymstyle{long-sc-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}
{
  \GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*\GenericAcronymFields{}%
\renewcommand*\acronymsort}[2]{##2}%
\renewcommand*\acronymentry}[1]{%
  \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glentryshort{##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
      {%
        \glsupcase
      }%
    }%
  }%
}

```

```

        \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
    \glsinsert
}%
{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
    \glsinsert
}%
{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
}%
}%
{%
    \glscapscase
    \glsinsert
    \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}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \glshyperfirstfalse
  \renewcommand*{\genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
    \protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
    \protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
    \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
    \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
  }%
  \renewcommand*{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
      (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
      (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsshortaccessdisplay
        {\mfirstucMakeUppercase
          {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
          (\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsshortpluralaccessdisplay

```

```

        {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glsentrylongpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
}
\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%

```

```

\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\newcommand*{\glsdefaultshortaccess}[2]{#1}
\renewcommand*{\newacronymhook}{%
  \edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%

```

```

description={\the\glslongtok},%
descriptionaccess=\relax,
text={\the\glsshorttok},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\noexpand\glslongaccessdisplay
  {\the\glslongtok}{\the\glslabeltok}\space
  (\noexpand\glsshortaccessdisplay
    {\the\glsshorttok}{\the\glslabeltok})},%
plural={\the\glsshorttok\noexpand\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*\showglongdescaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*\showglongdescpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*\showglongshortaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*\showglongshortpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*\showglonglongaccess}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*\showglolongpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}

```

8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
{%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
}%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
}%
}

```

8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2020/03/19 v4.46 (NLCT)]
\ifglsxindy
\renewcommand*\GlsAddXdyAttribute[1]{%
\edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\string~n\string\setentrycounter
{noexpand\glscounter}%
\expandafter\string\csname#1\endcsname
\expandafter@gobble\string\{\string" ^^J
:close \string"\expandafter@gobble\string\}\string" ^^J

```

```

:attr \string"#1\string"))}}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyperm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxindy
  \renewcommand*\GlsAddXdyLocation[2]{%
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\@xdyuserlocationnames{%
      \@xdyuserlocationnames^^J\space\space\space
      \string"#1\string"%
    }
  }
\fi
\renewcommand{\do@wrglossary}[1]{%
\ifglxindy
  \expandafter\@glo@check@mkidxrangear\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix)\relax
    \def\@glo@range{:close-range}%
  \fi

```

```

\fi
\protected@edef\@glslocoref{\theglsentrycounter}%
\@gls@checkmkidxchars\@glslocoref
\glossary[\csname glo@#1@type\endcsname]{%
(indexentry :tkey (\csname glo@#1@index\endcsname)
:locoref \string"\@glslocoref\string" %
:attr \string"\@gls@suffix\string" \@gls@range
)
}%
\else
\@set@gls@numformat\@gls@numfmt\@gls@counter\@gls@numberformat
\glossary[\csname glo@#1@type\endcsname]{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar\@gls@numfmt}{\theglsentrycounter}}%
\fi
}
\def\@set@gls@numformat#1#2#3{%
\expandafter\@gls@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@gls@prefix setentrycounter[] {#2}%
\expandafter\string\csname\@gls@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\ifglsxindy
\def\writeist{%
\openout\glswrite=\listfilename
\write\glswrite{;; xindy style file created by the glossaries
package in compatible-2.07 mode}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{ }(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%

```

```

\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-lowercase\string")
    :min-range-length \@glsminrange)}%
\else
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (:sep "\@gls@roman")
    :min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
  \string"Roman-page-numbers\string"^^J\space\space\space
  (\string"roman-numbers-uppercase\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-page-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"alpha-page-numbers\string"^^J\space\space\space
  (\string"alpha\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlpha compositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%

```

```

\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter\@gobble
  \string%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%

```

```

\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}

```

```

\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%

```

```

        {\glsinlinedescformat{##3}{##4}{##5}}%
\fi
\ifglschaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{##3}%
  \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
}
\compatglossarystyle{list}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
      \mbox{}\par\nobreak\@afterheading
      ##3\glspostdescription\space ##5}%
  \renewcommand{\glossarysubentryfield}[6]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%

```

```

\renewcommand*{\glossaryentryfield}[5]{%
  \item[]\makebox[\glslistdottedwidth][l]{%
    \glsentryitem{##1}\glstarget{##1}{##2}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
\renewcommand*{\glossarysubentryfield}[6]{%
  \item[]\makebox[\glslistdottedwidth][l]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{##3}%
    \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
}%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
  \csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
  \csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
  \csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%

```

```

\renewcommand*\glossaryentryfield}[5]{%
  \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]{%
    \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
  }%
  \compatglossarystyle{altlongragged4colheader}{%
    \csuse{@glscompstyle@altlong4col}%
  }%
  \compatglossarystyle{altlongragged4colborder}{%
    \csuse{@glscompstyle@altlong4col}%
  }%
  \compatglossarystyle{altlongragged4colheaderborder}{%
    \csuse{@glscompstyle@altlong4col}%
  }%
  \compatglossarystyle{index}{%
    \renewcommand*\glossaryentryfield}[5]{%
      \item\glstarget{##1}{\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
      \else
        % all other levels
        \subsubitem
      \fi
    }%
  }%

```

```

\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
\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
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else

```

```

        \space{##4}%
    \fi
    \space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
        \glssubentryitem{##2}%
    \fi
    \glstarget{##2}{\strut}%
    ##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
    \csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
    \csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
    \renewcommand{\glossaryentryfield}[5]{%
        \ifnum \@gls@prevlevel=0\relax
        \else
            \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
            \hangindent\glstreeindent
            \parindent\glstreeindent
        \fi
        \makebox[Opt][r]{\makebox[\glstreeindent][l]{%
            \glssubentryitem{##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{mcoltreename}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamegroup}{%
\csuse{@glscompstyle@tree}%
}

```

```

}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
  \csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
  \csuse{@glscompstyle@almtree}%
}%
}%

\compatglossarystyle{superragged}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##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]{%
    \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
}

```

```

\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \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@glstdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@glstdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@glstdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@glstdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\ifglshasprefix}[3]{%
\ifcempty{glo@glstdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\ifglshasprefixplural}[3]{%
\ifcempty{glo@glstdetoklabel{#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\@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}%
}

```

```

{%
  \ifglused{#2}%
  {%
    \ifglshasprefix{#2}{\glstryprefix{#2}\glsprefixsep}{}%
  }%
  {%
    \ifglshasprefixfirst{#2}{\glstryprefixfirst{#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}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefixplural{#2}{\glstryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirstplural{#2}%
      {\glstryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\PglS}{\@gls@hyp@opt\@PglS}
\newcommand*{\@PglS}[2][{}]{%
  \new@ifnextchar[%
    {\@PglS@{#1}{#2}}%
    {\@PglS@{#1}{#2}[{}]}%
}
\def\@PglS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglused{#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{\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}%
    {%
        \ifglused{#2}%
        {%
            \ifglshasprefixplural{#2}%
            {%
                \Glsentryprefixplural{#2}%
                \glsprefixsep
                \@glspl@{#1}{#2}[#3]%
            }%
            {\@Glspl@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirstplural{#2}%
            {%
                \Glsentryprefixfirstplural{#2}%
                \glsprefixsep
                \@glspl@{#1}{#2}[#3]%
            }%
            {\@Glspl@{#1}{#2}[#3]}%
        }%
    }%
}
}
\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][{}]{%
    \new@ifnextchar[%
        {\@PGLS@{#1}{#2}}%
        {\@PGLS@{#1}{#2}[{}]}%
    }
}

```

```

\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglsshasprefix{#2}%
      {\mfirstucMakeUppercase{\glsentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglsshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\@PGLSp1}
\newcommand*\@PGLSp1[2][]{%
  \new@ifnextchar[%
  {\@PGLSp1@{#1}{#2}}%
  {\@PGLSp1@{#1}{#2}[]}%
}
\def\@PGLSp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglsshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglsshasprefixfirstplural{#2}%
      {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{%
    }%
    \@GLSp1@{#1}{#2}[#3]%
  }%
}
}

```

8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2020/03/19 v4.46 (NLCT)]
\newcommand*\glsnavhyperlink[3][\@glo@type]{%
  \edef\gls@grplabel{#2}\protected@edef\@gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*\glsnavhyperlinkname[2]{\glsn:#1@#2}
\newcommand*\glsnavhypertarget[3][\@glo@type]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}

```

```

}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \@gls@target{\@glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }%
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type '#1'~^Jmissing group '#2'}%
    \gdef\@gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    \fi
  }
  \let\@gls@hypergroup@prerun\relax
  \AtEndDocument{\@gls@hypergroup@prerun}
  \newcommand*{\@gls@hypergroup}[2]{%
    \ifundefined{\@gls@hypergroup@list@#1}{%
      \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
    }{%
      \expandafter\let\expandafter\@gls@tmp
        \csname @gls@hypergroup@list@#1\endcsname
      \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
        \@gls@tmp,#2}%
    }%
  }
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{\@gls@hypergroup@list@\@gls@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergroup@list@\@gls@type\endcsname
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \@glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\@gls@hypernavsep
  }%
}
\newcommand*{\gls@hypernavsep}{\space\textbar\space}
\newcommand*{\gls@symbolnav}{%
  \@glsnavhyperlink{\gls@symbols}{\gls@getgrouptitle{\gls@symbols}}%
  \@gls@hypernavsep
}

```

```

\glsnavhyperlink{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%
    }%
  }%
  {%
    \ifglsdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglsdesc{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glsesubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{ }
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{ }
\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.10 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glslistgroupheaderfmt}[1]{#1}
\newcommand*{\glslistnavigationitem}[1]{\item[#1]}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\begin{description}}{\end{description}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}]%
      \glstarget{##1}{\glossentryname{##1}}%
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*{\subglossentry}[3]{%
    \glsesubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}

```

```

}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand{\subglossentry}[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*{\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}}%
  \renewcommand*{\subglossentry}[3]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}}%
}
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}

```

```

\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
}

```

8.11 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}{lp{\glsdescwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%

```

```

\renewcommand*\glossaryheader}{%
  \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{long3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}%
    {\end{longtable}}%
  \renewcommand*\glossaryheader}{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
  \renewcommand*\glsgroupskip}{}%
  \else
  \renewcommand*\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
    {\end{longtable}}%
  \renewcommand*\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*\glossaryheader}{%
    \hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
    }
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&

```

```

        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
}
\newglossarystyle{altlong4col}{%
    \setglossarystyle{long4col}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
        {\end{longtable}}}%
}
\newglossarystyle{altlong4colheader}{%
    \setglossarystyle{long4colheader}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
        {\end{longtable}}}%
}
\newglossarystyle{altlong4colborder}{%
    \setglossarystyle{long4colborder}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{|lp{\glstdescwidth}|lp{\glspagelistwidth}|}}%
        {\end{longtable}}}%
}
\newglossarystyle{altlong4colheaderborder}{%
    \setglossarystyle{long4colheaderborder}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{|lp{\glstdescwidth}|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}%
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%

```

```

        {\end{longtable}}%
    }
\newglossarystyle{altlongragged4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{altlong4col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glstdescwidth}l%
            >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}%
    }
\newcommand*{\glsLTpenaltycheck}{%
    \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
    \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
\renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
        \ifnum\outputpenalty > -\LT@end@pen
            \LT@err{floats and marginpars not allowed in a longtable}\@ehc
        \else
            \setbox\z@\vbox{\unvbox\@cclv}%
            \ifdim \ht\LT@lastfoot>\ht\LT@foot
                \dimen@\pagegoal
                \advance\dimen@-\ht\LT@lastfoot
                \ifdim\dimen@<\ht\z@
                    \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
                    \@makecol
                    \@outputpage
                    \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
                \fi
                \fi
            \global\@colroom\@colht
            \global\vsize\@colht
            {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
    \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsize\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
    \fi
}%
}

```

8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\ProvidesPackage{glossary-longragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}{%
    {\begin{longtable}[1>{\raggedright}p{glsdescwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}%
    \glspostdescription\space ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{glsdescwidth}|}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}
```

```

}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
    }
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l>\raggedright]p{\glsdescwidth}l%
      >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l>\raggedright]p{\glsdescwidth}l%
      >\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>\raggedright}p{\glsdescwidth}||l|%
      >\raggedright}p{\glspagelistwidth}||}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

        {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}
          >{\raggedright}p{\glspagelistwidth}|}}%
        {\end{longtable}}%
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}

```

8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.46]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}{%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}%
  }
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
  }
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
  }
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%

```

```

\renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
    \let\item\glstreeitem}%
  {\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
}

```

```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%

```

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
\setglossarystyle{mcolalmtree}%
\renewcommand*\glossaryheader{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%
\setglossarystyle{almtree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

8.15 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{glstdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{ }%
  \renewcommand*{\glsgroupheading}[1]{ }%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{ }%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}}%
    {\begin{supertabular}{|lp{glstdescwidth}|}}%
    {\end{supertabular}}%
}
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
      \bfseries\descriptionname\tabularnewline}}%
    {\tabletail{ }}%
    {\begin{supertabular}{lp{glstdescwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
      \bfseries \descriptionname\hline}}%
    {\tabletail{\hline}}
    {\begin{supertabular}{|lp{glstdescwidth}|}}%
}

```

```

        {\end{supertabular}}%
    }
\newglossarystyle{super3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}}%
    {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
     {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2} &
      ##3\tabularnewline
    }%
    \ifglsnogroupskip
      \renewcommand*{\glsgroupskip}{}%
    \else
      \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
    \fi
  }
\newglossarystyle{super3colborder}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
     {\end{supertabular}}%
  }
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
     \begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}%
     {\end{supertabular}}%
  }
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
     \tabletail{\hline}%
     \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
     {\end{supertabular}}%
  }
}

```

```

\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{1111}}{%
    \end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{1111}}{%
    \end{supertabular}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}}%
    {\end{supertabular}}%
}
\newglossarystyle{super4colheaderborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|1|1|1|1|}}%
    {\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuper4col}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{}}\tabletail{}%
    \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}%
    \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {llp{\glsdescwidth}llp{\glspagelistwidth}}%
    {\end{supertabular}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
  \setglossarystyle{super4colheaderborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {llp{\glsdescwidth}llp{\glspagelistwidth}}%
    {\end{supertabular}}%
}
}

```

8.16 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
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline} \tabletail{\hline}}%
    {\begin{supertabular}{||1>{\raggedright}p{\glsdescwidth}||}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
      \tabularnewline}%
      \tabletail{}}%
    {\begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
      \bfseries \descriptionname\tabularnewline\hline}}%
}

```

```

        \tabletail{\hline}
        \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||}%
        {\end{supertabular}}%
    }
\newglossarystyle{superragged3col}{%
\renewenvironment{theglossary}%
{\tablehead{\tabletail}{%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
>{\raggedright}p{\glspagelistwidth}||}%
{\end{supertabular}}%
}
\newglossarystyle{superragged3colheader}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline}\tabletail}{%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
}
\newglossarystyle{superragged3colheaderborder}{%
\setglossarystyle{superragged3colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
\renewenvironment{theglossary}%
{ \tablehead{} \tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{altsuperragged4colheader}{%
\setglossarystyle{altsuperragged4col}%
\renewenvironment{theglossary}%
{ \tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4colborder}{%
\setglossarystyle{altsuper4col}%
\renewenvironment{theglossary}%
{ \tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{ |l|>{\raggedright}p{\glsdescwidth}|l|%
>{\raggedright}p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {||>{\raggedright}p{\glsdescwidth}|||}%
      >{\raggedright}p{\glspagelistwidth}||}%
    {\end{supertabular}}%
}

```

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem
     \let\subitem\glstreesubitem
     \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%

```

```

        \item\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
        \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
    }%
\renewcommand{\subglossentry}[3]{%
    \ifcase##1\relax
        % level 0
        \item
    \or
        % level 1
        \subitem
        \glssubentryitem{##2}%
    \else
        % all other levels
        \subsubitem
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glsgroupheading}[1]{%
        \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
        \indexspace
    }%
}
\newglossarystyle{indexhypergroup}{%
    \setglossarystyle{index}%
    \renewcommand*{\glossaryheader}{%
        \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
    \renewcommand*{\glsgroupheading}[1]{%
        \item\glstreegroupheaderfmt
            {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
        \indexspace}%
}
\newglossarystyle{tree}{%
    \renewenvironment{theglossary}%
        {\setlength{\parindent}{0pt}%
         \setlength{\parskip}{0pt plus 0.3pt}}%
        {}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
        \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
        \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
    }%
}

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsgroupnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsgrouphypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi

```

```

\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsgnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsgnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glsgfindwidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forallglossaries[#1]{\@gls@type}%
{%
\forallglsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
}%
{%
\settothewidth{\dimen@}%
{\glstreenamefmt{\glsentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@glswidestname}{glo\@glsdetoklabel{\@glo@label}@name}%
\fi
}%
}%
}
\newcommand*{\glssetwidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][l]{#2}%
}
\newglossarystyle{almtree}{%

```

```

\renewenvironment{theglossary}%
  {\def\@gls@prevlevel{-1}%
   \mbox{}\par}%
  {\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
  \hangindent\glstreeindent
  \parindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
  \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength\glstreeindent\gls@tmplen
        \addtolength\glstreeindent\parindent
        \parindent\glstreeindent
      \else
        \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}{%
              \settowidth{\glstreeindent}{\glstreenamefmt{%
                \csname @glswidestname\romannumeral\@gls@prevlevel
                  \endcsname\space}}}%
          \addtolength\parindent{-\glstreeindent}%
          \setlength\glstreeindent\parindent
        \fi
        \fi
      \hangindent\glstreeindent
      \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
        \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
        \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
        \glossentrydesc{##2}\glspostdescription\space ##3\par
        \def\@gls@prevlevel{##1}%

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 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]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbolsymbol{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{%
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex){%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use

```

```

\string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
[\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
\ifcase\gls@seenoinde@nr
\renewcommand*\@gls@see@noindex{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key}%
}%
\or
\renewcommand*\@gls@see@noindex{%
\GlossariesWarning{'\gls@xr@key' key ignored}%
}%
\or
\renewcommand*\@gls@see@noindex{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
{\newcommand*\@glossarysec{section}}%
{\newcommand*\@glossarysec{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,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]{}
\glsentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*\@gls@counterwithin{#1}%
  \glsentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*\@gls@counterwithin{}
\newcommand*\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglsentrycounter
    \newcounter{glossarysubentry}[glossaryentry]%
    \else
    \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  }%
  \fi
  {}%
}

```

```

}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{%
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
\renewcommand*{\@glo@default@sorttype}{#1}%
\csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
\glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{%
\newcommand*{\@gls@setupsort@standard}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\ifx\@glo@sort\@glsdefaultsort
\let\@glo@sort\@glo@name
\fi
\let\glsdosanitizesort\@gls@sanitizesort
\glsprestandardsort{\@glo@sort}{##1}{##2}%
\expandafter\protected\xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected\xdef\csname glo@##2@sort\endcsname{%
\expandafter\glsortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}

```

```

}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@none{%
\def\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][1]{%
                \printglossary[type=\acronymtype,#1]}%
        }%
    \fi
}
\newcommand*{\@gls@do@acronymsdef}{%
\@gls@declareoption{acronyms}{%
    \glsacronymtrue
    \def\@gls@do@acronymsdef{%
        \DeclareAcronymList{acronym}%
        \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
        \renewcommand*{\acronymtype}{acronym}%
        \newcommand*{\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
}
\newcommand*{\@glsacronymlists}{%
\newcommand*{\@addtoacronymlists}[1]{%
    \ifx\@glsacronymlists\@empty
        \protected@xdef\@glsacronymlists{#1}%
    \else

```

```

        \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
    \fi
}
\newcommand*\DeclareAcronymList}[1]{%
    \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
    \protected@edef\@do@gls@islistofacronyms{%
        \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
    \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
    \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
        \def\@gls@before{##1}\def\@gls@after{##2}}%
    \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
    \ifx\@gls@after\@nnil
        #4%
    \else
        #3%
    \fi
}
}
\newif\if@gls@isacronymlist
\newcommand*\gls@checkisacronymlist}[1]{%
    \glsIfListOfAcronyms{#1}%
    {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\SetAcronymLists}[1]{%
    \renewcommand*\@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}%
}%
}

```

```

}
\glsssetexpandfield{type}
\glsssetnoexpandfield{desc}
\glsssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{\@gls@sanitizename}{}
\glsssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{\@gls@sanitizesymbol}{}
\glsssetnoexpandfield{symbol}
\glsssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{\@gls@sanitizesort}{%
  \ifglsssanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*\@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*\@gls@nosanitizesort{}
\newcommand*\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*\@gls@noidx@nosanitizesort{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@\@secondoftwo
  \let\@tabacckludge@\@secondoftwo
}

```

```

\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let'\@firstofone
\let'\@firstofone
\let^\@firstofone
\let"@@firstofone
\letu\@firstofone
\lett\@firstofone
\letd\@firstofone
\letr\@firstofone
\let=\@firstofone
\let.\@firstofone
\let~\@firstofone
\letv\@firstofone
\letH\@firstofone
\letc\@firstofone
\letb\@firstofone
\leta\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning[sanitize={description}] package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%

```

```

\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
  \GlossariesWarning{sanitize={name} package option deprecated}%
  \ifglS@sanitize@name
    \glSsetnoexpandfield{name}%
  \else
    \glSsetexpandfield{name}%
  \fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifglS@sanitize@symbol
    \glSsetnoexpandfield{symbol}%
    \glSsetnoexpandfield{symbolplural}%
  \else
    \glSsetexpandfield{symbol}%
    \glSsetexpandfield{symbolplural}%
  \fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesorttrue
      \glSsetnoexpandfield{sortvalue}%
    }%
  \else
    \glSsetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glSsanitizesortfalse
      \glSsetexpandfield{sortvalue}%
    }%
  \fi
}
\glSsanitizesorttrue
\glSsetnoexpandfield{sortvalue}%
\newcommand*{\@glS@noidx@setsanitizesort}{%
  \glSsanitizesortfalse
  \glSsetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
  \setbool{glSsanitizesort}{#1}%
  \ifglSsanitizesort
    \glSsetnoexpandfield{sortvalue}%
  \else
    \glSsetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}

```

```

}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glssetexpandfield{name}%
      \glssetexpandfield{symbol}%
      \glssetexpandfield{symbolplural}%
      \glssetexpandfield{desc}%
      \glssetexpandfield{descplural}%
    }%
  }%
  \setkeys{gls}{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{notranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
      \glstranslatetrue
    \fi
  }%
}

```

```

\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}{%

```

```

\glxindytrue
\gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘restoremakegloss’ has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been restored}%
\let\@domakeglossaries\@firstofone
}%
}%
}
\newcommand*{\@do@write@glslabels}{%
\AtEndDocument{\@do@write@glslabels}%
\let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
\newwrite\@gls@labelsfile
\immediate\openout\@gls@labelsfile=\jobname.glslabels

```

```

\protected@edef\@gls@fulllist{\@glo@types}%
\ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
\forallglsentries[\@gls@fulllist]{\@glsentry}%
{%
  \ifdefempty{\@glsentry}{}%
  {%
    \ifx\do@write@gls@labelsname@gobble
      \def\@gls@thisentryname{%
    \else
      \letcs{\@gls@thisentryname}{glo@\glsdetoklabel{\@glsentry}@name}%
      \@onelevel@sanitize\@gls@thisentryname
    \fi
    \immediate\write\@gls@labelsfile
      {\@glsentry\do@write@gls@labelsname{\@gls@thisentryname}}%
  }%
}%
\immediate\closeout\@gls@labelsfile
}
\newcommand*\do@write@gls@labelsname[1]{
\begingroup
\catcode'\^^I12
\gdef\@write@gls@labelsname#1{\^^I#1}
\endgroup
\@gls@declareoption{writeglslabels}{\do@write@gls@labels
\let\do@write@gls@labelsname@gobble}
\@gls@declareoption{writeglslabelnames}{\do@write@gls@labels
\let\do@write@gls@labelsname\@write@gls@labelsname}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*\@gls@doautomake{%
      \PackageError{glossaries}{You must use
        \string\makeglossaries\space with automake=true}
      {%
        Either remove the automake=true setting or
        add \string\makeglossaries\space to your document preamble.%
      }%
    }%
  \else
    \renewcommand*\@gls@doautomake{}%
  \fi
}

```

```

\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}{%
    {%
      \booltrue{glscompatible-3.07}%
    }%
  }%
}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
instead. If you \MessageBreak need the original definition of
\string\makeglossary\space use \MessageBreak the package options
kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
  \def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%

```

```

[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
  \ifcase\gls@debug@nr\relax
  \def\glossary{\@gls@override@glossary}%
  \def\makeglossary{%
    \GlossariesWarning{Use of \string\makeglossary\space with
      glossaries.sty is deprecated. Use \string\makeglossaries\space
      instead. If you need the original definition of
      \string\makeglossary\space use the package options
      kernelglossredefs=false (to prevent redefinition of
      \string\makeglossary) and nomain (if the file extensions cause a
      conflict)}%
    \makeglossaries
  }%
  \or
  \let\glossary\gls@original@glossary
  \let\makeglossary\gls@original@makeglossary
  \or
  \def\makeglossary{\makeglossaries}%
  \renewcommand*\@gls@override@glossary}[1][main]{%
    \gls@glossary{##1}%
  }%
  \fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*\@gls@do@symbolsdef{}
\newcommand*\@gls@symbolsdef{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*\@printsymbols}[1][\printglossary[type=symbols,##1]]{%
  \newcommand*\gls@tr@set@symbols@toctitle}{%
    \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
  }%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*\@gls@do@numbersdef{}
\newcommand*\@gls@numbersdef{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glnumbersgroupname}%
  \newcommand*\@printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
  \let\@gls@do@indexdef\@gls@indexdef
  \fi
}

```

```

}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]%
  \newcommand*{\newterm}[2][\%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \expandtwoargs
  \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
  \ifin@
  \@use@option
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
\fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.49]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
  \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
  \def\@gls@setupshortcuts{%
    \ifglsacrshortcuts
    \DefineAcronymSynonyms
    \fi
  }%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsentrycounter
  \let\@gls@doentrycounterdef\relax
\else
  \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter

```

```

\fi
\ifglssubentrycounter
  \let\@gls@dosingentrycounterdef\relax
\else
  \let\@gls@dosingentrycounterdef\@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@dosingentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
  \ifcsundef{chapter}{}%
  {%
    \let\@gls@old@chapter\@chapter
    \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
    \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
  }%
}%
{}
\newcommand*\@gls@onlypremakeg{}
\newcommand*\@onlypremakeg[1]{%
  \ifx\@gls@onlypremakeg\@empty
    \def\@gls@onlypremakeg{#1}%
  \else
    \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
    \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
  \fi
}
\newcommand*\@disable@onlypremakeg{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
  \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*\@disable@premakecs[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
  used before \string\makeglossaries}{You can't use
  \string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle[1]{%
  \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}

```

```

\providecommand*\symbolname}{Symbol}
\providecommand*\pagelistname}{Page List}
\providecommand*\glssymbolsgroupname}{Symbols}
\providecommand*\glsnumbersgroupname}{Numbers}
\newcommand*\glspluralsuffix}{s}
\newcommand*\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix}{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename}{see}
\providecommand*\andname}{\&}
\newcommand*\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}%
}
\newcommand*\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp
            \renewcommand*\glossaryname}{\translate{Glossary}}%
          }%
          \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
        }%
      }%
    }%
  }%
}
\ifcsundef{captions#1}{%
}
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
  }
}

```

```

    { .ldf }%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
  }{%
\fi
\DeclareRobustCommand*{\nopostdesc}{%
\newcommand*{\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
  \ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi

```

```

}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{}
\newcommand*\glscompositor{.}
\newcommand*\glsSetCompositor[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*\gls@suffixF{}
\newcommand*\glsSetSuffixF[1]{%
  \renewcommand*\gls@suffixF{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*\gls@suffixFF{}
\newcommand*\glsSetSuffixFF[1]{%
  \renewcommand*\gls@suffixFF{#1}}
}
\ifcsundef{hyperlink}%
{
  \newcommand*\glsnumberformat[1]{#1}%
}%
{
  \newcommand*\glsnumberformat[1]{\glshypernumber{#1}}%
}
\newcommand*\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}%
}%
{%
  \@pglossarysection{#1}{#2}%
}%
\glsglossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\glsglossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }
  {%
    \newcommand{\glsglossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
  \else
    \@mkboth{#1}{#1}%
  \fi
}
\newcommand*{\setglossarysection}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossarysection}[2]{%
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname[#1]{#2}%
  }%
  {%
    \csname\@glossarysec\endcsname*{#2}%
  }%
}

```

```

    \@gls@toc{#1}{\@glossarysec}%
  }%
  \@glossaryseclabel
}
\newcommand*\@p@glossarysection}[2]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%
    \@gls@toc{#1}{\@glossarysec}%
    \csname\@glossarysec\endcsname*{#2}%
  }%
  \@glossaryseclabel
}
\newcommand*\gls@docclearpage}{%
  \ifthenelse{\equal{\@glossarysec}{chapter}}{%
    {%
      \ifcsundef{cleardoublepage}%
      {%
        \clearpage
      }%
      {%
        \ifcsdef{if@openright}%
        {%
          \if@openright
            \cleardoublepage
          \else
            \clearpage
          \fi
        }%
        {%
          \cleardoublepage
        }%
      }%
    }%
  }%
}
\newcommand*\glsclearpage}{\gls@docclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
  \ifglsnumberline
  \addcontentsline{toc}{#2}{\protect\numberline{}#1}%
  \else
  \addcontentsline{toc}{#2}{#1}%
  \fi
\fi
}

```

```

\newcommand*\glsnoxindywarning}[1]{%
  \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning}[1]{%
  \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
  \edef\xdyattributes{\string"default\string"%}
\fi
\ifglsxindy
  \edef\xdyattributelist{}%
\fi
\ifglsxindy
  \def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\do@ifinlist##1,#1,##2\end@ifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \do@ifinlist,#2,#1,\end@ifinlist
}
\ifglsxindy
  \newcommand*\@xdycounters{\glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
      \do@addcounter
    }
  }
  \@onlypremakeg\GlsAddXdyCounters
\else
  \newcommand*\GlsAddXdyCounters[1]{%
    \glsnoxindywarning\GlsAddXdyAttribute
  }
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first

```

```

instance of \string\GlsAddXdyAttribute}%
}
\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@glsaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxindywarning\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{hyperarm}
\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{}
\ifglsexindy
  \newcommand*\@GlsAddXdyAlphabet}[2]{%
  \edef\@xdyuseralphabets{%
    \@xdyuseralphabets ^~J
    (define-alphabet "#1" (#2))}}
\else
  \newcommand*\@GlsAddXdyAlphabet}[2]{%
    \glsnoxindywarning\@GlsAddXdyAlphabet}
\fi
\ifglsexindy
  \newcommand*\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{

```

```

        :sep \string"\@gls@roman\string"%
    }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
    \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
    \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
    \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
    \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
    \string"ALPHA\string"
    :sep \string"\@gls@Alphacompositor\string"
    \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
    \string"arabic-numbers\string"
    :sep \string"\glscompositor\string"
    \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][ ]{%
    \def\@gls@tmp{#1}%
    \ifx\@gls@tmp\@empty
        \edef\@xdyuserlocationdefs{%
            \@xdyuserlocationdefs ^^J%
            (define-location-class \string"#2\string"^^J\space\space
            \space(:sep \string"{}\glsopenbrace\string" #3
            :sep \string"\glsclosebrace\string"))
        }%
    \else
        \edef\@xdyuserlocationdefs{%
            \@xdyuserlocationdefs ^^J%
            (define-location-class \string"#2\string"^^J\space\space
            \space(:sep "\glsopenbrace"
            #1
            :sep "\glsclosebrace\glsopenbrace" #3
            :sep "\glsclosebrace"))
        }%
    \fi
}

```

```

\fi
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation}[2]{%
    \glsnoxywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
  \string"roman-page-numbers\string"^^J\space\space\space
  \string"arabic-page-numbers\string"^^J\space\space\space
  \string"arabic-section-numbers\string"^^J\space\space\space
  \string"alpha-page-numbers\string"^^J\space\space\space
  \string"Roman-page-numbers\string"^^J\space\space\space
  \string"Alpha-page-numbers\string"^^J\space\space\space
  \string"Appendix-page-numbers\string"
  \@xdyuserlocationnames^^J\space\space\space
  \string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
  \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoxywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
  \expandafter\toks@\expandafter{\@xdysortrules}%
  \protected@edef\@xdysortrules{\the\toks@ ^^J
    (sort-rule \string"#1\string" \string"#2\string")}%
}
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
\def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
\newcommand*\GlsAddXdyStyle[1]{%
  \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else

```

```

\newcommand*\GlsAddXdyStyle[1]{%
  \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage@empty
      \@ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifgls@xindy@glsnumbers
    \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

```

```

\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
  }%
\newcommand*\forallglossaries}[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms}[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{#3}%
  }%
}
\newcommand*\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forglsentries[\@@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{glo@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*\ifglsused}[3]{%
  \ifbool{glo@glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%

```

```

    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
    been defined}{-}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
    has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
    has not been defined}{You need to define a glossary entry before you
    can use it.}%
    #3%
  }%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{-}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@gl@sentries\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]{%
  \ifcsequal{glo@glstetoklabel{#1}@desc}{%
    {#3}%
    {#2}%
  }%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@glstetoklabel{#1}@desc}{@no@post@desc}{%
    {#2}%
    {#3}%
  }%
}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@glstetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
    {#3}%
    {%
      \ifdefequal\@glo@symbol\@gls@default@value
        {#3}%
        {#2}%
      }%
    }%
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@glstetoklabel{#1}@long}%
  \ifdefempty\@glo@long
    {#3}%
    {%
      \ifdefequal\@glo@long\@gls@default@value
        {#3}%
        {#2}%
      }%
    }%
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glstetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
    {%
      \ifdefequal\@glo@short\@gls@default@value
        {#3}%
        {#2}%
      }%
    }%
}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
{%
  \ifdefequal\@glo@thisvalue\@gls@default@value
  {%
    #4%
  }%
  {%
    \let\glscurrentfieldvalue\@glo@thisvalue
    #3%
  }%
}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo\@glsdetoklabel{#2}\@gls@thisfield}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
    {%
      \ifdefequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
      {%
        \let\glscurrentfieldvalue\@glo@thisvalue
        #3%
      }%
    }%
  }%
  }%
  {%
    \GlossariesWarning{Unknown entry field '#1'}%
    #4%
  }%
}%
}%
}
\newcommand*\@glscurrentfieldvalue{}
\newcommand*\@glo@types}{,}
\newcommand*\@gls@provide@newglossary{%
  \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*\@defglsentryfmt}[2][\@glsdefaulttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*\@gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtext}[1]{%

```

```

\ifboolexpr{test {\ifdefstring{#1}{tex}}
            or test {\ifdefstring{#1}{TEX}}}
{%
  \def#1{nottex}%
  \PackageError{glossaries}%
  {Forbidden '.tex' extension replaced with '.nottex'}%
  {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
   Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }{%
      \toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
      \expandafter\gdef\csname glolist@#2\endcsname{,}%
      \expandafter\edef\csname @glo@type@#2@log\endcsname{#1}%
      \expandafter\edef\csname @glo@type@#2@in\endcsname{#3}%
      \expandafter\edef\csname @glo@type@#2@out\endcsname{#4}%
      \expandafter\@gls@forbidtextext\csname @glo@type@#2@log\endcsname
      \expandafter\@gls@forbidtextext\csname @glo@type@#2@in\endcsname
      \expandafter\@gls@forbidtextext\csname @glo@type@#2@out\endcsname
      \expandafter\def\csname @glo@type@#2@title\endcsname{#5}%
      \@gls@provide@newglossary
      \protected@write\@auxout{\string\@newglossary{#2}{#1}{#3}{#4}}%
      \ifcsundef{gls@#2@entryfmt}%
      {%
        \defglsentryfmt[#2]{\glsentryfmt}%
      }%
      }%
    }%
    \@gls@defsortcount{#2}%
    \@ifnextchar[{\@gls@setcounter{#2}}%
      {\@gls@setcounter{#2}[\glscounter]}%
  }%
  }%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%

```

```

\newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
\expandafter\def\csname @glo@type@#1@counter\endcsname{#2}%
\ifglsxindy
\GlsAddXdyCounters{#2}%
\fi
}
\newcommand*\@gls@getcounter}[1]{%
\csname @glo@type@#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{glo@list@#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@ig@type{#1}%
\expandafter\DTLifinlist\expandafter
{\@gls@ig@type}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}

```

```

\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key[3]{%
\renewcommand*\@gls@xr@key{#1}%
\gls@checkseeallowed
\def#2{#3}%
}

```

```

    \@glo@seeautonumberlist
  }
  \newcommand*{\gls@xr@key}{see}
  \newcommand*{\gls@checkseeallowed}{%
    \@gls@see@noindex
  }
  \newcommand*{\gls@checkseeallowed@preambleonly}{%
    \GlossariesWarning
    {'\gls@xr@key' key doesn't have any effect when used in the document
    environment. Move the definition to the preamble
    after \string\makeglossaries\space
    or \string\makenoidxglossaries}%
  }
  \define@key{glossentry}{parent}{%
    \def\@glo@parent{#1}}
  \define@choicekey{glossentry}{nonumberlist}{%
    [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]}%
  {%
    \ifcase\gls@nonumberlist@nr\relax
      \def\@glo@prefix{\glsnonextpages}%
      \@gls@savenonumberlist{true}%
    \else
      \def\@glo@prefix{\glsnextpages}%
      \@gls@savenonumberlist{false}%
    \fi
  }
  \newcommand*{\@gls@savenonumberlist}[1]{%
  \newcommand*{\@gls@initnonumberlist}{}%
  \newcommand*{\@gls@storenonumberlist}[1]{%
  \newcommand*{\@gls@enablesavenonumberlist}{%
    \renewcommand*{\@gls@initnonumberlist}{%
      \undef\@glo@nonumberlist
    }%
    \renewcommand*{\@gls@savenonumberlist}[1]{%
      \def\@glo@nonumberlist{##1}%
    }%
    \renewcommand*{\@gls@storenonumberlist}[1]{%
      \ifdef\@glo@nonumberlist
      {%
        \cslet{glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
      }%
      {}%
    }%
    \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
  }
  \define@key{glossentry}{user1}{%
    \def\@glo@useri{#1}%
  }
  \define@key{glossentry}{user2}{%
    \def\@glo@userii{#1}%
  }

```

```

}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*{\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@glsdefaultplural}{}
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
  Maybe makeglossaries + rerun required}%
\else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
  }%
}

```

```

        to reference location lists.%
    }%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
}%
}

```

```

        {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
        }%
    }%
}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\@gls@expandfields{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\@gls@noexpandfields{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%
{%
\gls@defglossaryentry{#1}{#2}%
}%
}
\newcommand*\@gls@defdocnewglossaryentry{%
\let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
\ifglsentryexists{#1}%
{}%
{%
\gls@defglossaryentry{#1}{#2}%
}%
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\@gls@glsdefs@labellist{}
\newcommand*\@gls@glsdefs@addlabel}[1]{\listxadd{\@gls@glsdefs@labellist}{#1}}
\newcommand*\@gls@docdef@labellist{}
\newcommand*\@gls@docdef@addlabel}[1]{\listxadd{\@gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
\ifundef\@gls@deffile

```

```

{%
  \global\newwrite\@gls@deffile
  \immediate\openout\@gls@deffile=\jobname.glsdefs
}%
{}%
\ifglsentryexists{#1}%
{%
  \xifinlist{#1}{\gls@docdef@labellist}%
  {%
    \GlossariesWarning{Entry ‘#1’ has already been defined in the
      document environment (ignoring new definition)}%
  }%
  {%
    \xifinlist{#1}{\gls@glsdefs@labellist}%
    {%
      \gls@defglossaryentry{#1}{#2}%
      \gls@docdef@addlabel{#1}%
      \@gls@writedef{#1}%
    }%
    {%
      \GlossariesWarning{Entry ‘#1’ has already been defined in
        the preamble (in which case remove the duplicate definition)
        or you have just updated to glossaries.sty
        v4.47 (in which case delete ‘\jobname.glsdefs’)}%
    }%
  }%
}%
{}%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode‘\noexpand\@=\number\catcode‘\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{\relax}{\relax}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*\gls@sdefs@newdocentry}[2]{%
  \ifglsentryexists{#1}%
  {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
    (if you have moved the definition there from the document
    environment, delete \jobname.glsdefs)}}%
  {}%
}

```

```

    \gls@glsdefs@addlabel{#1}%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}@\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
        \glspercentchar
      }%
    }%
  }%
  {}%
}%
\gls@writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%

```

```

{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
      {%
        \edef#1{\expandafter\@secondoftwo\@gls@map}%
        \@endfortrue
      }%
    }%
  }%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  }%
  }%
  \@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{-%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  }%
  \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%

```

```

{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname gls@assign@#1@field\endcsname}[2]{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}%
{}%
\@gls@saddkey{#1}%
}
\newcommand*{\@gls@saddkey}[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{\glsfieldfdef}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\glo@label @#2}%
  {%
    \letcs#3{glo@\glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname

```

```

\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
\fi
}
\newcommand{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\ifglsfielddfeq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\expandafter\ifdefstrequal
\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand*{\gls.writedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\@glo@desc}%
\gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%

```

```

{%
  \bgroup
  \let\@org@newglossaryentryprehook\@newglossaryentryprehook
  \long\def\@newglossaryentryprehook{%
    \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
    \@org@newglossaryentryprehook
  }%
  \renewcommand*\@gls@assign@desc}[1]{%
    \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
    \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
  }
  \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@gls@name
  \let\@glo@desc\@gls@desc
  \let\@glo@descplural\@gls@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
  \def\@glo@parent{}%
  \def\@glo@prefix{}%
  \@gls@initnonumberlist
  \def\@glo@useri{}%
  \def\@glo@userii{}%
  \def\@glo@useriii{}%
  \def\@glo@useriv{}%
  \def\@glo@userv{}%
  \def\@glo@uservi{}%
  \def\@glo@short{}%
  \def\@glo@shortpl{}%
  \def\@glo@long{}%
}

```

```

\def\@glo@longpl{}%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glolist@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
{}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\empty
  \fi
}%
{}%
{}%
{}%
\protected@edef\@glolist@\csname glolist@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
  \@glolist@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
    }%
  }%
\fi

```

```

\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
{%
\PackageError{glossaries}%
{%
Invalid parent ‘\@glo@parent’
for entry ‘\@glo@label’ - parent doesn’t exist%
}%
{%
Parent entries must be defined before their children%
}%
\def\@glo@parent{}%
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
{\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname}%
{\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @plural\endcsname}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\else
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glo@type@\@glo@type @counter}%
{%
\def\@glo@defaultcounter{\glscounter}%
}%
{%
\letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%

```

```

\gls@assign@field{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
  \@glsnoname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
  \@glsnodesc
  \let\@glo@desc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
  \gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\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@glsee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
      \do@glsee
    }%
    \@glo@autoseehook
  }%
}
\newcommand*{\@glo@autoseehook}{%
}
\newcommand*{\@newglossaryentryprehook}{%
}
\newcommand*{\@newglossaryentryposthook}{%
}
\newcommand*{\@newglossaryentry@defcounters}{%
}
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\glo@list{,%
}
\for@glseentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\@glo@thislabel\glo@label
      {}{\protected@eappto\glo@list{\glo@label,}}%
    }%
  }
\cslet{glo@list@\glo@type}{\glo@list}%
\csdef{glo@\@glo@thislabel @type}{#2}%
}
\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}%
  \ifglsxindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\@glo@parent\@empty
      \@onelevel@sanitize\@glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossaryentryfield{\@glo@esclabel}%
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
      }%
    \fi
  \fi
}
\AtBeginDocument{%
  \ifpackageloaded{amsmath}%
  {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
  {}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
  \ifmeasuring@
  \else
  #1%
  \fi
}

```

```

\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\gls@patchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\gls@patchtabularx\relax
  }%
  {}%
}
\newcommand*\gls@reset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@gls@reset{#1}%
    }%
  }%
}
\newcommand*\gls@localreset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@gls@localreset{#1}%
    }%
  }%
}
\newcommand*\gls@unset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@gls@unset{#1}%
    }%
  }%
}
\newcommand*\gls@localunset[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@gls@localunset{#1}%
    }%
  }%
}
\newcommand*\@gls@localunset{\@gls@localunset}
\newcommand*\@gls@localunset[1]{%

```

```

    \expandafter\let\csname ifglo@glsetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
    \expandafter\global\csname glo@glsetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
    \expandafter\let\csname ifglo@glsetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
    \expandafter\global\csname glo@glsetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glsreset{\@glsentry}%
    }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glslocalreset{\@glsentry}%
    }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glsunset{\@glsentry}%
    }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glslocalunset{\@glsentry}%
    }%
}
\newcommand*\@newglossaryentry@defcounters{%
    \csdef{glo@\@glo@label @currcount}{0}%
    \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*\glsenableentrycount{%
    \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
    \renewcommand*\gls@defdocnewglossaryentry{%
        \renewcommand*newglossaryentry[2]{%
            \PackageError{glossaries}{\string\newglossaryentry\space
                may only be used in the preamble when entry counting has
                been activated}{If you use \string\glsenableentrycount\space
                you must place all entry definitions in the preamble not in

```

```

        the document environment}%
    }%
}%
\newcommand*\glsentrycurrcount}[1]{%
  \ifcsundef{glo@glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
  \ifcsundef{glo@glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \csdef{glo@glsdetoklabel{##1}@currcount}{0}%
}%
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl@plformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@pl@{##1}{##2}[##3]%
  \fi
}%
\def\@cGl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gl@{##1}{##2}[##3]%
  \fi
}

```

```

}%
\def\cGlspl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \cGlspl@{##1}{##2}[##3]%
  \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
    {}%
  }%
}
\newcommand*{\@gls@entry@count}[2]{}
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2] []{%
  \new@ifnextchar[{\@cgl@{##1}{##2}}{\@cgl@{##1}{##2} []}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl}\space is defaulting to
  \string\gls\space since you haven't enabled entry counting}%
  \@cgl@{##1}{##2}[##3]%
}
\newcommand*{\cglformat}[2]{%
  \ifglshaslong{##1}{\glsentrylong{##1}}{\glsentryfirst{##1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2] []{%
  \new@ifnextchar[{\@cGls@{##1}{##2}}{\@cGls@{##1}{##2} []}%
}

```

```

}
\def\cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@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*\glsformat}[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}%
  }%
}%
}%
{%
\ifglused\glslabel
{%
  \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
  {\glsentrydescplural{\glslabel}}%

```

```

        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    }%
    {%
        \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    }%
} %
} %
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}}%
}%
{%
#1{\glsentryfirst{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\ifbool{glscompatible-3.07}%
{%
\protected@edef\@glo@etext{%
#2{\glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}}%
\xmakefirstuc\@glo@etext
}%
{%
#2{\Glsentrytext{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}}%
}%
}%
\ifbool{glscompatible-3.07}%
{%
\protected@edef\@glo@etext{%
#1{\glsentryfirst{\glslabel}}%
{\glsentrydesc{\glslabel}}%
{\glsentrysymbol{\glslabel}}{\glsinsert}}%
\xmakefirstuc\@glo@etext
}%
} %
} %

```

```

        {%
            #1{\Glsentryfirst{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}%
        }%
    }%
}%
{%
    \ifglsused\glslabel
    {%
        \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
    {%
        \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
}
}%
}%
{%
    \ifglsused{\glslabel}%
    {%
        #2{\glscustomtext}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{%
    }%
    {%
        #1{\glscustomtext}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{%
    }%
}
}%
}
\newcommand*{\glsgenentryfmt}{%
    \ifdefempty\glscustomtext
    {%
        \glsifplural
        {%
            \glscapscase
            {%
                \ifglsused\glslabel
                {%
                    \glsentryplural{\glslabel}\glsinsert
                }%
                {%
                    \glsentryfirstplural{\glslabel}\glsinsert
                }%
            }
        }
    }
}

```

```

}%
{%
  \ifglused\glslabel
  {%
    \Glsentryplural{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirstplural{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \mfirstucMakeUppercase
      {\glsentryplural{\glslabel}\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
      {\glsentryfirstplural{\glslabel}\glsinsert}%
  }%
}%
}%
}%
\glscapscase
{%
  \ifglused\glslabel
  {%
    \glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
}%
{%
  \ifglused\glslabel
  {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
  }%
}%

```



```

        {%
        \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][\glsdefaultttype]{%
    \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{\gls\type}{\@gls@nohyperlist}%
\KV@glslink@hyperfalse}{}%
}
\newcommand*\@gls@setdefault@glslink@opts{}
\def\@gls@link[#1]#2#3{%
\leavevmode
\protected@edef\glslabel{\glsdetoklabel{#2}}%
\def\@gls@link@opts{#1}%
\let\@gls@link@label\glslabel
\def\@glsnumberformat{glsnumberformat}%
\protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
\protected@edef\gls\type{\csname glo@\glslabel @type\endcsname}%
\let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
\@gls@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{\glslinkprefix\glslabel}{\gls\textformat{#3}}%
\else
\glsdonohyperlink{\glslinkprefix\glslabel}{\gls\textformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*\glslinkprefix{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*\@gls@saveentrycounter{%
\def\@gls@Hcounter{}%
\ifthenelse{\equal{\@gls@counter}{equation}}%
{
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}
}
}

```

```

\else
  \savecounters@
  \advance\c@equation by 1\relax
  \edef\theglentrycounter{\csname the\@gls@counter\endcsname}%
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\@gls@Hcounter{\theglentrycounter}%
  }%
  {%
    \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
  }%
  \protected@edef\theHglentrycounter{\@gls@Hcounter}%
  \restorecounters@
\fi
}%
}%
{%
  \edef\theglentrycounter{\expandafter\noexpand
    \csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
  \ifcsundef{theH\@gls@counter}%
  {%
    \def\theHglentrycounter{\theglentrycounter}%
  }%
  {%
    \protected@edef\theHglentrycounter{\expandafter\noexpand
      \csname theH\@gls@counter\endcsname}%
  }%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
  \expandafter\@glo@check@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@suffi{x}{glsnumberformat}%
\else
\def\@glo@suffi{x}{#2}%
\fi
\else
\def\@glo@prefix{ }\def\@glo@suffi{x}{#1#2}%
\fi
\fi}
\newcommand*\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostubst
}%
\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@checkesbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescllevel#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@checkesquote#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@checkeschar{\@gls@checkeschar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@@gls@checkeschar{\@gls@checkeschar#2\|\|\null}%
\else
\def\@@gls@checkeschar{\@gls@checkeschar#2|#3\null}%
\fi
\fi
\@gls@checkeschar
}
\def\@gls@checkescllevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescllevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@@gls@checkescllevel{\@gls@checkescllevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!\!\null}%
\else
\def\@@gls@checkescllevel{\@gls@checkescllevel#2\!#3\null}%
\fi
\fi
\@gls@checkescllevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@=#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

        \gls@quotechar\gls@encapchar\gls@quotechar\gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@levelchar\gls@quotechar\gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2|#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\gls@quotechar\gls@actualchar\gls@quotechar\gls@actualchar}%
\def\@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else

```

```

\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesc{|relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar|@backslashchar|\@gls@quotechar|@backslashchar}%
\def\@gls@checkesc{|\@gls@checkesc#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar|@backslashchar}%
\ifx\null#3\null
\def\@gls@checkesc{|\@gls@checkesc#2\\ \null}%
\else
\def\@gls@checkesc{|\@gls@checkesc#2\#3\null}%
\fi
\fi
|\@gls@checkesc
}
\egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@xdycheckquote{relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
|string"\string"}%
\def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
\fi
\else

```

```

\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"}%
\ifx\null#3\null
\def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
\@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\gls@scaps\@firstofthree
\let\gls@customtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\gls@type @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
\glslocalunset{#2}%
}

```

```

        \else
        \glsunset{#2}%
        \fi
    }%
    \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][{}]{%
    \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]]}%
}
\def\@Gls@#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
        \let\glsifplural\@secondoftwo
        \let\gls caps case\@secondofthree
        \let\gls custom text\@empty
        \def\glsinsert{#3}%
        \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
        \@gls@link[#1]{#2}{\@glo@text}%
        \ifKV@glslink@local
            \glslocalunset{#2}%
        \else
            \glsunset{#2}%
        \fi
    }%
    \glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\newcommand*{\@GLS}[2][{}]{%
    \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]]}%
}
\def\@GLS@#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
        \let\glsifplural\@secondoftwo
        \let\gls caps case\@thirdofthree
        \let\gls custom text\@empty
        \def\glsinsert{#3}%
        \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
        \@gls@link[#1]{#2}{\@glo@text}%
        \ifKV@glslink@local
            \glslocalunset{#2}%
        \else
            \glsunset{#2}%
        \fi
    }%
    \glspostlinkhook
}

```

```

\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] []{%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\gls insert{#3}%
    \def\@glo@text{\csname gls@\gls type @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@gls link@local
      \gls local unset{#2}%
    \else
      \gls unset{#2}%
    \fi
  }%
  \gls post link hook
}
\newrobustcmd*{\GLspl}{\@gls@hyp@opt\@GLspl}
\newcommand*{\@GLspl}[2] []{%
  \new@ifnextchar[{\@GLspl@{#1}{#2}}{\@GLspl@{#1}{#2} []}%
}
\def\@GLspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@secondofthree
    \let\gls custom text\@empty
    \def\gls insert{#3}%
    \def\@glo@text{\csname gls@\gls type @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@gls link@local
      \gls local unset{#2}%
    \else
      \gls unset{#2}%
    \fi
  }%
  \gls post link hook
}
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\newcommand*{\@GLSpl}[2] []{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2} []}%
}
\def\@GLSpl@#1#2[#3]{%
  \glsdoifexists{#2}%

```

```

{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\gl@ifplural\@firstoftwo
  \let\gl@scapscase\@thirdofthree
  \let\gl@customtext\@empty
  \def\gl@insert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \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@ifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else
      \gl@unset{#2}%
    \fi
  }%
  \gl@postlinkhook
}
\newcommand*{\@gl@link@nocheckfirsthyper}{}
\newcommand*{\@gl@field@link}[3]{%
  \gl@doifexists{#2}{%
    {%
      \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
      \@gl@link[#1]{#2}{#3}%
    }%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@stext}{\@gl@hyp@opt\@gl@stext}
\newcommand*{\@gl@stext}[2][{}]{%
  \new@ifnextchar[{\@gl@stext@{#1}{#2}}{\@gl@stext@{#1}{#2}[]}}
\def\@gl@stext@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\gl@entrytext{#2}#3}%
}
\newrobustcmd*{\GL@stext}{\@gl@hyp@opt\@GL@stext}

```

```

\newcommand*{\@GLStext}[2] [] {%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} [] }
\def\@GLStext@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2] [] {%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} [] }
\def\@GLStext@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2] [] {%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} [] }
\def\@glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2] [] {%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} [] }
\def\@Glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2] [] {%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} [] }
\def\@GLSfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} [] }
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} [] }
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} [] }
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} [] }

```

```

\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glentryfirstplural{#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}{\glentryfirstplural{#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{\glentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2][{}]{%
  \new@ifnextchar[{\@glsname@{#1}#{2}}{\@glsname@{#1}#{2}[]}]
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2][{}]{%
  \new@ifnextchar[{\@Glsname@{#1}#{2}}{\@Glsname@{#1}#{2}[]}]
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2][{}]{%
  \new@ifnextchar[{\@GLSname@{#1}#{2}}{\@GLSname@{#1}#{2}[]}]
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\mfirstucMakeUppercase{\glentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2][{}]{%
  \new@ifnextchar[{\@glsdesc@{#1}#{2}}{\@glsdesc@{#1}#{2}[]}]
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\newcommand*{\@Glsdesc}[2][{}]{%
  \new@ifnextchar[{\@Glsdesc@{#1}#{2}}{\@Glsdesc@{#1}#{2}[]}]
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2][{}]{%
  \new@ifnextchar[{\@GLSdesc@{#1}#{2}}{\@GLSdesc@{#1}#{2}[]}]
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\mfirstucMakeUppercase{\glentrydesc{#2}#3}}%
}

```

```

}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\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*{\glsymbol}{\@gls@hyp@opt\@glsymbol}
\newcommand*{\@glsymbol}[2] [] {%
  \new@ifnextchar[{\@glsymbol@{#1}{#2}}{\@glsymbol@{#1}{#2} []}]
\def\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\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*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
\newcommand*{\@glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}]
\def\@glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}

```

```

\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar [{\@GLSsymbolplural@#1}{#2}]{\@GLSsymbolplural@{#1}{#2} []}}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar [{\@glsuseri@{#1}{#2}]{\@glsuseri@{#1}{#2} []}}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2] [] {%
  \new@ifnextchar [{\@Glsuseri@{#1}{#2}]{\@Glsuseri@{#1}{#2} []}}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar [{\@GLSuseri@{#1}{#2}]{\@GLSuseri@{#1}{#2} []}}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
  \new@ifnextchar [{\@glsuserii@{#1}{#2}]{\@glsuserii@{#1}{#2} []}}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar [{\@Glsuserii@{#1}{#2}]{\@Glsuserii@{#1}{#2} []}}
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar [{\@GLSuserii@{#1}{#2}]{\@GLSuserii@{#1}{#2} []}}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar [{\@glsuseriii@{#1}{#2}]{\@glsuseriii@{#1}{#2} []}}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar [{\@Glsuseriii@{#1}{#2}]{\@Glsuseriii@{#1}{#2} []}}

```

```

\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*\@Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*\@Glsuseriii}[2][]{%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}]
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*\@glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*\@glsuseriv}[2][]{%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2}[]}]
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*\@Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*\@Glsuseriv}[2][]{%
  \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}]
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\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}{\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\gls@scaps@case\@firstofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \gls@postlinkhook
}
\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\gls@label{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls@scaps@case\@secondofthree
    \let\glsinsert\@empty
    \def\gls@customtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \gls@postlinkhook
}
}

```

```

\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@\gls@type @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@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2] [] {%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
  }%
}

```

```

\def\glscustomtext{%
  \acronymfont{\Glsentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][{}]{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}%
}
\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \def\glslabel{#2}%
  \let\glsifplural\@secondoftwo
  \let\gls caps case\@secondofthree
  \let\glsinsert\@empty
  \def\gls custom text{%
    \Glsentrylong{#2}#3%
  }%
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\gls post link hook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\@ns@ACRlong}
\newcommand*{\ns@ACRlong}[2][{}]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@thirdofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\@ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
}

```

```

\glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@glsdetoklabel{#1}@#2}%
  }%
}

```

```

\ifdef\@glo@text
{%
  \xmakefirstuc{\@glo@text}%
}%
{%
  ??\PackageError{glossaries}{The field '#2' doesn't exist for glossary
  entry '\glsdetoklabel{#1}'}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%
{%
  ??%
}%
}
\newcommand*\@Gls@entryname}[1]{\@Gls@entry@field{#1}{name}}
\newrobustcmd*\Gls@entryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*\@Gls@acronymname}[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*\glentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
  \newrobustcmd*\Glsentrydesc}[1]{%
    \@Gls@entry@field{#1}{desc}%
  }
  \newcommand*\glentrydescplural}[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}%
  }
  \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}%
{
\begin{group}
\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
\end{group}
}%
}
}
\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\glsrbrace{\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] {%
    \glsnoxindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%

```

```

    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglxindy
  \def\writeist{%
    \ifundef{\glswrite}{\newwrite\glswrite}{}%
    \@gls@addpredefinedattributes
    \openout\glswrite=\istfilename
    \write\glswrite{;; xindy style file created by the glossaries
      package}%
    \write\glswrite{;; for document '\jobname' on
      \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\@xdystyle:=\@xdyrequiredstyles\do{%
      \ifx\@xdystyle\@empty
      \else
        \protected@write\glswrite{{(require
          \string"\@xdystyle.xdy\string")}%
        }%
      \fi
    }%
    \write\glswrite{^^J%

```

```

; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{{(define-location-class
    \string"\@gls@classI\string"^^J\space\space\space
    (
      :sep "{{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%

```

```

\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite-{}{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glspercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty

```

```

\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
  '\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}

```

```

\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@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@checkeschar####1\noexpand\@nil
\noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@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}%

```



```

\else
  \expandafter\newwrite\csname glo@#1@file\endcsname
  \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
  \fi
  \@gls@renewglossary
  \writeist
}%
{%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
    \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
\ifglsxindy
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\@empty}%
    \ifcsundef{@xdy@#1@language}%
    {\let\@gls@langmod\@xdy@main@language}%
    {\letcs\@gls@langmod{@xdy@#1@language}}%
    \edef\@gls@dothiswrite{\noexpand\write18{xindy
      -I xindy
      \@gls@order
      -L \@gls@langmod\space
      -M \gls@istfilebase\space
      -C \gls@codepage\space
      -t \jobname.\csuse{@glotype@#1@log}
      -o \jobname.\csuse{@glotype@#1@in}
      \jobname.\csuse{@glotype@#1@out}}}%
    }%
    \@gls@dothiswrite
  }%
  {%
    \GlossariesWarning{Can’t make glossary ‘#1’, it doesn’t exist}%
  }%
}
\else
\newcommand*{\@gls@automake}[1]{%

```

```

\ifglossaryexists{#1}
{%
  \@closegls{#1}%
  \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-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
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
    {%
      \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-M ord/letorder }}%
        {\let\@gls@order\@empty}%
      \ifcsundef{@xdy@#1@language}%
        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{@xdy@#1@language}}%
      \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
        -I xindy
        \@gls@order
        -L \@gls@langmod\space
        -M \gls@istfilebase\space
        -C \gls@codepage\space
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}}%
      }%
      \@gls@dothiswrite
    }%
    {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
      doesn't exist. Rerun may be required}}%
  }%
  {%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
  }%
}
}

```

```

\else
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@gloftype@#1@out}}{%
      {%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-1 }}%
        {\let\@gls@order\@empty}%
        \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
          -s \istfilename\space
          -t \jobname.\csuse{@gloftype@#1@log}
          -o \jobname.\csuse{@gloftype@#1@in}
          \jobname.\csuse{@gloftype@#1@out}}}%
        }%
        \@gls@dothiswrite
      }%
      {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gloftype@#1@out}
        doesn't exist. Rerun may be required}}%
    }%
    {\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
  }%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand*\@gls@@automake@immediate}{%
  \ifnum\gls@automake@nr=2\relax
  \@for\@gls@type:=\@glo@types\do{%
    \ifdefempty{\@gls@type}{}%
    {\@gls@automake@immediate{\@gls@type}}%
  }%
  \glsautomakefalse
  \renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
  \@domakeglossaries
  {%
    \@gls@@automake@immediate
    \ifundef{\glswrite}{\newwrite\glswrite}{}%
    \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}}
    \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}}
    \ifundef\@gls@extramakeindexopts
    {}%
    {%
      \protected@write\@auxout{}{\string\providecommand
        \string\@gls@extramakeindexopts[1]{}}
      \protected@write\@auxout{}{\string\@gls@extramakeindexopts

```

```

    {\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}\string\@glsorder{\glsorder}}
\@for\@glo@type:=\@glo@types\do{%
  \ifthenelse{\equal{\@glo@type}{}}{}{%
    \makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4] []{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglsavenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*\@printgloss@setsort{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
  \renewcommand*\@gls@doautomake{%
    \@for\@gls@type:=\@glo@types\do{%
      \ifdefempty{\@gls@type}{}%
        {\@gls@automake{\@gls@type}}%
      }%
    }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not

```

```

        have a glossary}%
    }%
}%
\AtEndDocument{%
    \warn@nomakeglossaries
    \warn@noprintglossary
}
\newcommand*\makenoidxglossaries{%
    \@domakeglossaries
    {%
        \renewcommand{\@gls@nofwarn}[1]{%
            \GlossariesWarning{Empty glossary for
            \string\printnoidxglossary[type={##1}].
            Rerun may be required (or you may have forgotten to use
            commands like \string\gls)}%
        }%
        \let\@gls@checkmkidxchars\@gobble
        \glsesclocationsfalse
        \let\@do@wrglossary\gls@noidxglossary
        \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
        \let\@gls@checkseeallowed\relax
        \renewcommand{\@do@seeglossary}[2]{%
            \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@glsetoklabel{##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@glsetoklabel{##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@glsetoklabel{##1}@loclist}%
\let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
\let\@gls@org@glsseeformat\glsseeformat
\let\glsnoidxdisplayloc##2\relax

```

```

\let\glsseeformat##3\relax
\ifdef\@gls@loclist
{%
  \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
\let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
\let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@glslocref}%
    }%
  }%
}

```

```

    }%
  }
  \providecommand\istfile{\glswrite}
  \AtEndDocument{%
    \glswritefiles
  }
  \newcommand*\@glswritefiles{%
    \forallglossaries{\@glo@type}{%
      \ifcsundef{glo@\@glo@type @filetok}%
        {%
          \def\gls@tmp{}}%
        }%
        {%
          \edef\gls@tmp{\expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        }%
        \ifx\gls@tmp\@empty
          \ifx\@glo@type\glsdefaulttype
            \GlossariesWarningNoLine{Glossary '@@glo@type' has no
              entries.^^JRemember to use package option 'nomain' if
you
              don't want to^^Juse the main glossary}%
          \else
            \GlossariesWarningNoLine{Glossary '@@glo@type' has no
              entries}%
          \fi
        \else
          \@glsopenfile{\glswrite}{\@glo@type}%
          \immediate\write\glswrite{%
            \expandafter\the
              \csname glo@\@glo@type @filetok\endcsname}%
          \immediate\closeout\glswrite
        \fi
      }%
    }
  \newcommand*\@gls@glossary}[1]{%
    \@gls@glossary{#1}%
  }
  \newcommand*\@gls@glossary}[2]{%
    \if@gls@debug
      \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
  }
  \newcommand*\@gls@renewglossary{%
    \gdef\@gls@glossary##1{\@bsphack\beginingroup\gls#wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
  }
  \newcommand*\@gls#wrglossary}[2]{%
    \ifgls@savewrites
      \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \fi
  }

```

```

\expandafter\global\expandafter\csname glo@#1@filetok\endcsname
\expandafter{\@gls@tmp^~J}%
\else
\ifcsdef{glo@#1@file}%
{%
\expandafter\protected@write\csname glo@#1@file\endcsname{%
\gls@disablepagerefexpansion}{#2}%
}%
{%
\ifignoredglossary{#1}{}%
{%
\GlossariesWarning{No file defined for glossary '#1'}%
}%
}%
\fi
\endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
\glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*\glswriteentry[2]{%
\ifglsindexonlyfirst
\ifglsused{#1}{#2}%
\else
#2%
\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*\gls@disablepagerefexpansion{%
\@for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*\gls@alphpage}{\@alph\c@page}
\newcommand*\gls@Alphpage}{\@Alph\c@page}
\newcommand*\gls@numberpage}{\number\c@page}
\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\gls@saddprotectedpagefmt}[1]{%
\protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
\csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
\protected@eappto\@wrglossarynumberhook{%
\noexpand\let\expandonce{\csname org@gls#1\endcsname}%
\expandonce{\csname#1\endcsname}%
\noexpand\def\expandonce{\csname#1\endcsname}{%
\noexpand\@wrglossary@pageformat
\expandonce{\csname gls#1page\endcsname}}%
}

```

```

        \expandonce{\csname org@gls#1\endcsname}%
    }%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@@@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
      \def\number{\expandafter\gls@number}%
    \fi
    \def\@arabic##1{%
      \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%

```

```

\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
  \def\@glo@counterprefix{%
\else
  \protected@edef\@glsHlocref{\theglsentrycounter}%
  \@gls@checkmkidxchars\@glsHlocref
  \edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
    {\@glslocref}{\@glsHlocref}%
  }%
  \@do@gls@getcounterprefix
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@@wrglossary
}
\newcommand*{\@do@@wrglossary}{%
  \ifglsxindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"{\@glo@counterprefix}{\@glslocref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
  \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
  {\@glo@counterprefix}%
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@glslocref}}%

```

```

\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
\edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been^^J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:tkey (\csname glo@#1@index\endcsname)
:xref (\string"\@gls@xref\string")
:attr \string"see\string"
)
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{#{#2#3}}%
\fi
}
}

```

```

\def\@gls@fixbraces#1[#2]#3\end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat[3][\seename]{%
  \emph{#1} \glsseeitem{#2}}
\newrobustcmd*\glsseeelist[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseeelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseeelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat[1]{\glsentrytext{#1}}
\newcommand*\gls@save@numberlist[1]{%
  \ifglssavenumberlist
    \toks@{#1}%
    \protected@edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}{\@do@writeaux@info}%
  \fi
}
\newcommand*\warn@noprntglossary{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored}[2]{%
  \ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}

```

```

}%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary ‘\@glo@type’. \MessageBreak
        Using deprecated fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
  {%
    \ifx\glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type @title\endcsname
        \glossarytitle

```

```

\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers@glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotoc@title
\@glossarystyle
\let\gls@org@glossaryentryfield@glossentry
\let\gls@org@glossarysubentryfield@subglossentry
\renewcommand{\glossentry}[1]{%
  \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
  \gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
  \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
  \gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\egroup
\global\let@glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \input@{\jobname.\csname @glo@type\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type\@glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglsxindy
  \ifcsundef{@xdy@\@glo@type @language}%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdy@language[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdy@language{\@glo@type}{\@xdy@main@language}%
        }%
      }%
    }%
  }%
  }%
  {%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdy@language[2]{}%
        }%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\@xdy@language{\@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@glstdetoklabel{#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}{@glstref@\@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@glstdetoklabel{#1}@sort}%
    \letcs@gls@sort@B{glo@glstdetoklabel{#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@glstdetoklabel{#1}@sort}%
    \letcs@gls@sort@B{glo@glstdetoklabel{#2}@sort}%
    \edef@glo@do@compare{%
        \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce@gls@sort@B}%
    }%
}

```

```

        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
}
\newcommand*\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{ }%
  }%
}
}
\newcommand*\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{ }%
  }%
}
}
\newcommand*\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@\glsorder}%

```

```

    {%
      \@glo@sortentries{\csuse{@glo@sorthandler@\glsorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glsorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=nocase]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglsentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listead{\@glo@sortinglist}{\@gls@thislabel}%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%

```

```

{}%
{%
  \listcsadd{@glsref@\glo@type}{#1}%
}%
\ifcsdef{@glo@sortingchildren@#1}%
{%
  \@glo@addchildren{\glo@type}{#1}%
}%
{}%
}
\newcommand*{\@glo@sortmacro@use}[1]{
\newcommand*{\@print@noidx@glossary}{%
\ifcsdef{@glsref@\glo@type}%
{%
  \ifcsdef{@glo@sortmacro@\glo@sorttype}%
  {%
    \csuse{@glo@sortmacro@\glo@sorttype}{\@glo@type}%
  }%
  {%
    \PackageError{glossaries}{Unknown sort handler '\@glo@sorttype'}{}%
  }%
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \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]{%
\ifglsonumberlist
  \def\glossaryentrynumbers##1{#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]{%
  \ifglssentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*\glsentrycounterlabel}{%
  \ifglssentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*\glsentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}
\newcommand*\glsentryitem}[1]{%
  \ifglssentrycounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*\glsentrysubitem}[1]{%
  \ifglssubentrycounter
    \glsstepsubentry{#1}\glsentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{
  \newenvironment{theglossary}{}{}%
}%
{
  \@gls@warnontheGLOSSDEFINED
  \renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}

```

```

\newcommand*\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\providecommand*\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}}%
    {\the\toks@}%
  }%
  \@do@glossentry
}
\newcommand*\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\newcommand*\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\newcommand*\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\providecommand*\compatiblesubglossentry}[3]{%

```

```

\toks@{#3}%
\protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
{#2}%
  {\noexpand\glsnamefont
  {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
  {\expandafter\expandonce\csname glo@#2@desc\endcsname}}%
  {\expandafter\expandonce\csname glo@#2@symbol\endcsname}}%
  {\the\toks@}%
}%
\@do@subglossentry
}
\newcommand*\setglossentrycompatibility{%
\let\glossentry\compatibleglossentry
\let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand{\glossaryentryfield}[5]{%
\GlossariesWarning
{Deprecated use of \string\glossaryentryfield.^^J
I recommend you change to \string\glossentry.^^J
If you've just upgraded, try removing your gls auxiliary
files^^J and recompile}%
\noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*\glossarysubentryfield}[6]{%
\GlossariesWarning
{Deprecated use of \string\glossarysubentryfield.^^J
I recommend you change to \string\subglossentry.^^J
If you've just upgraded, try removing your gls auxiliary
files^^J and recompile}%
\glstarget{#2}{\strut}#4. #6\par}
\newcommand*\glsgroupskip{}
\newcommand*\glsgroupheading[1]{}
\newcommand*\glsgetgrouptitle[1]{%
\@gls@getgrouptitle{#1}{\@gls@grptitle}%
\@gls@grptitle
}
\newcommand*\@gls@getgrouptitle[2]{%
\dtl@ifsingle{#1}%
{%
\ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
}%
{%
\ifboolexpr{test{\ifstrequal{#1}{glsymbols}}
or test{\ifstrequal{#1}{glsnumbers}}}%
{%
\ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
}%
{%
\def#2{#1}%
}%
}

```

```

}%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\newcommand*{\glsgetgrouplabel}[1]{%
\ifthenelse{equal{#1}{\glsymbolsgroupname}}{\glsymbols}{%
\ifthenelse{equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*{\glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
    I recommend you switch to \string\setglossarystyle\space unless
    you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
      {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax

```

```

    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\@glsnumber#1\nohyperpage{}}\@nil}
}
\def\@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
  \@glsnumberlink{#1}\delimR\@glsnumberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}

```

```

\def\@delimN#1\delimN #2\delimN#3\{\%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
    \hyperlink{\glsentrycounter\@glo@counterprefix\the\toks@}%
      {\the\toks@}%
  \fi
\else
  \@gls@ReturnAfterFi{%
    \@gls@removespaces#2\@nil
  }%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand\oldacronym[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%

```

```

        \noexpand\gls{#1}\noexpand\xspace}%
    }%
}
\newcommand{\newacronym}[4][[]]{
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*ns@acrfull[2][[]]{%
\new@ifnextchar[{\@acrfull{#1}{#2}}%
{\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
\acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullfmt}[3]{%
\acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
\acrfullformat{#1}{#3}{#4}{#5}{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace(#2)}
\newrobustcmd*{\glspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*ns@Acrfull[2][[]]{%
\new@ifnextchar[{\@Acrfull{#1}{#2}}%
{\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
\Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
\acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*ns@ACRfull[2][[]]{%
\new@ifnextchar[{\@ACRfull{#1}{#2}}%
{\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
\ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
\acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}

```

```

\newcommand*\ns@acrfullpl[2][1]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[1]}%
}
\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][1]{%
  \new@ifnextchar[{\@AcRfullpl{#1}{#2}}%
    {\@AcRfullpl{#1}{#2}[1]}%
}
\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][1]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[1]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand*\ACRfullplfmt[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acrnameformat[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand*\newacronym[4][1]{%
    \ifdefempty{\@glsacronymlists}%
      {%
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
      }%
    }%
}

```

```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\protected@edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\expandonce{\acronymentry{##2}}},%
    sort={\acronymssort{\the\glsshorttok}{\the\glslongtok}},%
    text={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    \GenericAcronymFields,%
    \the\glskeylisttok
  }%
}%
\do@newglossaryentry
}%
\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*\GenericAcronymFields{description={\the\glslongtok}}
\newcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*\acronymssort}[2]{##1}
\newcommand*\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@##1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘##1’}{%
  }%
  }%
}

```

```

\ifdefempty{\@glsacronymlists}%
{%
  \DeclareAcronymList{\acronymstype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
}%
}%
}
\newcommand*\@newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
\newcommand*\@renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
  }%
}
\newcommand*\@GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\@GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\@GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\@genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\@Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\@genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space

```

```

(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\space
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
  \glsentrylong{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
  \Glsentrylong{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%

```

```

\renewcommand*\genacrfullformat}[2]{%
  \protect\firstacronymfont{\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}%
{%
  \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}%

```

```

{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%

```

```

    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
      }%
    }%
  }%
}

```

```

    {%
      \mfirstucMakeUppercase
        {\glsentrylongpl{\glslabel}\glsinsert}%
    }%
  }%
  {%
    \glscapscase
    {%
      \glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \Glsentrylong{\glslabel}\glsinsert
    }%
    {%
      \mfirstucMakeUppercase
        {\glsentrylong{\glslabel}\glsinsert}%
    }%
  }%
  }%
  {%
    \glsgenentryfmt
  }%
}
{\glscustomtext\glsinsert}%
}
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}

```

```

\renewcommand*\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
  \renewcommand*\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \glshyperfirstfalse
  \renewcommand*\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2%
    \protect\footnote{\glsentrylong{##1}}%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \firstacronymfont{\Glsentryshort{##1}}##2%
    \protect\footnote{\glsentrylong{##1}}%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
    \protect\footnote{\glsentrylongpl{##1}}%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
    \protect\footnote{\glsentrylongpl{##1}}%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
    (\glsentrylong{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
    (\glsentrylong{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
      (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%

```

```

\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\acrlong
  \let\Acl\Acrlong
  \let\aclp\acrlongpl
  \let\Acpl\Acrlongpl
  \let\acf\acrfull
  \let\Acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}

```

```

}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
  }%
  \do@newglossaryentry
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@symbolplural\@org@gls@assign@symbolplural
  \@@gls@noexpand@assign@default@plurals
}
\newcommand*{\@@gls@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\glslabeltok @plural\endcsname\@@gls@default@acr@plural
    \csxdef{glo@the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
  \fi

```

```

\expandafter\ifx\csname glo@the\glslabeltok @firstpl\endcsname\@gls@@default@acr@firstplural
\csxdef{glo@the\glslabeltok @firstpl}{%
\expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
\fi
\expandafter\ifx\csname glo@the\glslabeltok @descplural\endcsname\@gls@@default@acr@descplural
\csxdef{glo@the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
\fi
}
\newcommand*{\@gls@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@gls@@default@acr@firstplural}{%
\expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
}
\newcommand*{\@gls@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDefaultAcronymDisplayStyle{\@gls@type}%
}%
}
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDefaultAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink[#1]{#2}{#3}}%
}
\newcommand*{\acrnohreffootnote}[3]{%
\footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\def\glsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\glsentryfmt}%
}%
}%
\firstacronymfont{\glsentryfmt}%
}

```

```

\ifglshsymbol{\glslabel}%
{%
  \expandafter\protect\expandafter\acrfootnote\expandafter
  {\@gls@link@opts}{\@gls@link@label}%
  {%
    \glsifplural
      {\glsentrysymbolplural{\glslabel}}%
      {\glsentrysymbol{\glslabel}}%
    }%
  }%
}%
}%
}{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    symbol={\the\glslongtok},%
    symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl

```

```

\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{footnote,smallcaps,description}%
{footnote-sc-desc}{short-sc-footnote-desc}%
\else
\ifglsacrsmaller
\@gls@deprecated@acr@warning{footnote,smaller,description}%
{footnote-sm-desc}{short-sm-footnote-desc}%
\else
\@gls@deprecated@acr@warning{footnote,description}%
{footnote-desc}{short-footnote-desc}%
\fi
\fi
\renewcommand{\newacronym}[4][[]]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionFootnoteNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand*{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
\ifglsacrdua
\PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
\defglsentryfmt[##1]{\glsgenentryfmt}%
}
\newcommand*{\DescriptionDUANewAcronymDef}{%
\edef\@do@newglossaryentry{%

```

```

\newcommand{\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}%
  \fi
}

```

```

\DeclareAcronymList{\@glo@type}%
\SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@gls@acronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*\@SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert@empty
\acronymfont{\gls@genentryfmt}\gls@org@insert
}%
{%
\gls@genentryfmt
\ifgls@has@symbol{\glslabel}%
{%
\gls@if@plural
{%
\def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
}%
{%
\def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
}%
\space(\protect\firstacronymfont
{\gls@scaps@case
{\@glo@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
}%
}%
\ifgls@customtext\glsinsert}%
}%
}
\newcommand*\@DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%

```

```

{%
  type=\acronymtype,%
  name={\noexpand
    \acronymformat{\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:=\@gls@acronymlists\do{%
    \SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
    \renewcommand{\acronymfont}[1]{\textsc{##1}}
    \renewcommand*\{acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
    \ifglsacrsmaller
        \renewcommand*\{acronymfont}[1]{\textsmaller{##1}}%
    \fi
\fi
\fi
}%
\newcommand*\{SetFootnoteAcronymDisplayStyle}[1]{%
    \defglsentryfmt[##1]{%
        \ifdefempty\glscustomtext
        {%
            \let\gls@org@insert\glsinsert
            \let\glsinsert\@empty
            \ifglsused{\glslabel}%
            {%
                \acronymfont{\glsentryfmt}\gls@org@insert
            }%
            {%
                \firstacronymfont{\glsentryfmt}\gls@org@insert
                \ifglsashaslong{\glslabel}%
                {%
                    \expandafter\protect\expandafter\acrfootnote\expandafter
                    {\@gls@link@opts}{\@gls@link@label}%
                }%
                \glsifplural
                {\glsentrylongpl{\glslabel}}%
                {\glsentrylong{\glslabel}}%
            }%
        }%
        }%
    }%
    {\glscustomtext\glsinsert}%
}%

```

```

}
\newcommand*{\FootnoteNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      first={\the\glsshorttok},%
      firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%
    \@gls@expand@field{##1}{descplural}{##2}%
  }%
  \do@newglossaryentry
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*{\SetFootnoteAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
    \else
      \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
    \fi
  \fi
  \renewcommand{\newacronym}[4] []{%
    \ifx\@glsacronymlists\@empty

```

```

\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetFootnoteAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\FooterNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand*\@acronymfont}[1]{\textsc{##1}}%
\renewcommand*\@acrpluralsuffix{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*\@acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
\ifglsacrdua
\PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\DeclareRobustCommand*\glsdoparenifnotempty}[2]{%
\protected@edef\gls@tmp{##1}%
\ifdefempty\gls@tmp
{}%
{}%
\ifx\gls@tmp\@gls@default@value
\else
\space (#2{##1})%
\fi
}%
}
\newcommand*\@SetSmallAcronymDisplayStyle}[1]{%
\defglsentryfmt[##1]{%
\ifdefempty\gls@customtext
{}%
\let\gls@org@insert\gls@insert
\let\gls@insert\@empty
\ifglsused{\gls@label}%
{}%
\acronymfont{\gls@genentryfmt}\gls@org@insert
}%
{}%

```

```

\glsgenentryfmt
\ifglshassymbol{\glslabel}%
{%
  \glsifplural
  {%
    \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
  }%
  {%
    \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
  }%
  \space
  (\glscapscase
  {\firstacronymfont{\@glo@symbol}}%
  {\firstacronymfont{\@glo@symbol}}%
  {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
{}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\SmallNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
  type=\acronymtype,%
  name={\noexpand\acronymfont{\the\glsshorttok}},%
  sort={\the\glsshorttok},%
  text={\the\glsshorttok},%
  plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
  first={\the\glslongtok},%
  firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
  short={\the\glsshorttok},%
  shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
  long={\the\glslongtok},%
  longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
  description={\noexpand\@glo@first},%
  descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
  symbol={\the\glsshorttok},%
  symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
  \the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}

```

```

}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*\@SetSmallAcronymStyle{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
  \else
    \@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
  \fi
  \renewcommand{\newacronym}[4][1]{%
    \ifx\@gls@acronymlists\@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:=\@gls@acronymlists\do{%
    \SetSmallAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*\@acronymfont[1]{\textsc{##1}}
    \renewcommand*\@acrpluralsuffix{\glsupacrpluralsuffix}%
  \else
    \renewcommand*\@acronymfont[1]{\textsmaller{##1}}
  \fi
  \ifglsacrdua
    \ifglsacrsmallcaps
      \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
        can’t both be set}{}%
    \else

```

```

        \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
\fi
}%
\newcommand*{\SetDUADisplayStyle}[1]{%
    \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\DUANewAcronymDef}{%
    \edef\do@newglossaryentry{%
        \noexpand\newglossaryentry{\the\glslabeltok}%
        {%
            type=\acronymtype,%
            name={\the\glsshorttok},%
            text={\the\glslongtok},%
            first={\the\glslongtok},%
            plural={\noexpand\expandonce\noexpand\@glo@longpl},%
            firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
            short={\the\glsshorttok},%
            shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
            long={\the\glslongtok},%
            longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
            description={\the\glslongtok},%
            descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
            symbol={\the\glsshorttok},%
            symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
            \the\glskeylisttok
        }%
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@gls@expand@field{##1}{symbolplural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
\let\gls@assign@descplural\@org@gls@assign@descplural

```

```

}
\newcommand*{\SetDUASStyle}{%
  \@gls@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDUADisplayStyle{\@gls@type}%
  }%
}
\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifglsacrdescription
    \ifglsacrfootnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifglsacrdua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \else
    \ifglsacrfootnote
      \SetFootnoteAcronymStyle
    \else
      \ifthenelse{\boolean{glsacrsmalldcaps}\OR
        \boolean{glsacrsmalld}}{
        %
        \SetSmallAcronymStyle
      }%
    \fi
    \ifglsacrdua

```

```

        \SetDUASStyle
        \fi
    }%
    \fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
    \def\glsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
    name={\the\glsshorttok},%
    description={\the\glslongtok},%
    first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
    firstplural={\acrfullformat
        {\noexpand\glsentrylongpl{\the\glslabeltok}}%
        {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
    text={\the\glsshorttok},%
    plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
    \protected@edef\@do@newglossaryentry{%
        \noexpand\newglossaryentry{\the\glslabeltok}%
        {%
            type=\acronymtype,%
            short={\the\glsshorttok},%
            shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
            long={\the\glslongtok},%
            longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
            user1={\the\glsshorttok},%
            user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
            user3={\the\glslongtok},%
            user4={\the\glslongtok\noexpand\acrpluralsuffix},%
            \CustomAcronymFields,%
            \the\glskeylisttok
        }%
    }%
    \@do@newglossaryentry
}
\newcommand*{\SetCustomStyle}{%
    \renewcommand{\newacronym}[4][[]]{%
        \ifx\@glsacronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetCustomDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
    }%
}

```

```

    \glslongtok{##4}%
    \newacronymhook
    \CustomNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetCustomDisplayStyle{\@gls@type}%
  }%
}
\RequirePackage{glossary-hypernav}[=v4.49]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
}

```

```

\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*\showgloname}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*\showglodesc}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*\showglodescplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*\showglosort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*\showglosymbol}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*\showgloshort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*\showgloloclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \cshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*\showglossaries}{%
  \show\@glo@types
}

```

```

}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx\@glsxtr@doaccsupp\empty
  \GlossariesWarning{The ‘glossaries-accsupp’
    package has been loaded\MessageBreak
    after the ‘glossaries-extra’ package. This\MessageBreak
    can cause a failure to integrate both packages. \MessageBreak
    Either use the ‘accsupp’ option when you load\MessageBreak
    ‘glossaries-extra’ or load ‘glossaries-accsupp’\MessageBreak
    before loading ‘glossaries-extra’}%
  \fi
}
{}
\def\compatibleglossentry#1#2{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{%
    \noexpand\accsuppglossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
    {\the\toks@}%

```

```

}%
\do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@glstetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@glstetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@glstetoklabel{#2}@symbol\endcsname}%
{\the\toks@}}%
}%
\do@subglossentry
}
\RequirePackage{glossaries}[=v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\glsaccessibility}[4][ ]{%
\@glsshowaccsupp{#1}{#2}{#3}%
\gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
\def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
\def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%
\def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%

```

```

\def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
\def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
\def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
\def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
\def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
\def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
\def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
\def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
}
\appto\@gls@keymap{,%
{access}{access},%
{textaccess}{textaccess},%
{firstaccess}{firstaccess},%
{pluralaccess}{pluralaccess},%
{firstpluralaccess}{firstpluralaccess},%
{symbolaccess}{symbolaccess},%
{symbolpluralaccess}{symbolpluralaccess},%
{descaccess}{descaccess},%
{descpluralaccess}{descpluralaccess},%
{shortaccess}{shortaccess},%
{shortpluralaccess}{shortpluralaccess},%

```

```

{longaccess}{longaccess},%
{longpluralaccess}{longpluralaccess},%
{user1access}{useriaccess},%
{user2access}{useriiaccess},%
{user3access}{useriiiaccess},%
{user4access}{userivaccess},%
{user5access}{uservaccess},%
{user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*\@newglossaryentryprehook{%
\@gls@oldnewglossaryentryprehook
\def\@glo@access{\relax}%
\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
\def\@glo@longpluralaccess{\@glo@longaccess}%
\def\@glo@useriaccess{\relax}%
\def\@glo@useriiaccess{\relax}%
\def\@glo@useriiiaccess{\relax}%
\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*\@newglossaryentryposthook{%
\@gls@oldnewglossaryentryposthook
\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
\@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
\@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
\@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
\@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%

```

```

        \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaccess\endcsname{%
    \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaccess\endcsname{%
    \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
    \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
    \@glo@userviaccess}%
}
\newcommand*{\glsentryaccess}[1]{%
    \@gls@entry@field{#1}{access}%
}
\newcommand*{\glsentrytextaccess}[1]{%
    \@gls@entry@field{#1}{textaccess}%
}
}

```

```

\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}

```

```

}
\newcommand{\glsfieldaccsup}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsup}%
    {\csname glsxtr\glscategory{#4}#3accsup\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsup}%
      {\csname glsxtr\glscategory{#4}accsup\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsup}%
        {\csname gls#3accsup\endcsname{#1}{#2}}%
        {\glsaccsup{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsup}%
    {\csname gls#3accsup\endcsname{#1}{#2}}%
    {\glsaccsup{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsup}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsup\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsup}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsup}{\glsshortaccsup}
\newcommand*{\glsaccsup}[2]{\glsaccessibility{ActualText}{#1}{#2}}
\newcommand*{\xglsaccsup}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsup\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
  #1%
  \else
  \xglsaccsup{\@glo@access}{#1}%
  \fi
}
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
  {#3}%
  {\expandafter\glsfieldaccsup\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@access}%
  {#1}%
}

```

```

    {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
  }
  \newrobustcmd*{\glsstextaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
  }
  \newrobustcmd*{\glspluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
  }
  \newrobustcmd*{\glsfirstaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
  }
  \newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
  }
  \newrobustcmd*{\glsymbolaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
  }
  \newrobustcmd*{\glsymbolpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
  }
  \newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@descaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
  }
  \newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@descpluralaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
  }
  \newrobustcmd*{\glsshortaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@shortaccess}%
      {#1}%
    {\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
  }
  \newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\glsdetoklabel{#2}@shortpluralaccess}%
      {#1}%

```

```

    {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
  }
\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@longpluralaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@useriiiaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userivaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
}
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@uservaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@userviaccess}%
  {#1}%
  {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
  {%
    \PackageError{glossaries-accsupp}{No accessibility support
      for key ‘#1’}{%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
  }%
}

```

```

}%
}
\renewcommand*{\@@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glspluralaccessdisplay
            {\glsentryplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
              {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glsymbolpluralaccessdisplay
              {\glsentrysymbolplural{\glslabel}}{\glslabel}}
            {\glsinsert}}%
          }%
          {%
            #1{\glsfirstpluralaccessdisplay
              {\glsentryfirstplural{\glslabel}}{\glslabel}}%
              {\glsdescriptionpluralaccessdisplay
                {\glsentrydescplural{\glslabel}}{\glslabel}}%
              {\glsymbolpluralaccessdisplay
                {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
              {\glsinsert}}%
            }%
          }%
        }%
      }%
    }%
  }%
  \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
  {%
    \MakeUppercase{%
      #2{\glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}}%
      {\glsdescriptionpluralaccessdisplay
        {\glsentrydescplural{\glslabel}}{\glslabel}}%
      {\glsymbolpluralaccessdisplay
        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
    }%
  }%
  \MakeUppercase{%
    #1{\glsfirstpluralaccessdisplay
      {\glsentryfirstplural{\glslabel}}{\glslabel}}%
      {\glsdescriptionpluralaccessdisplay
        {\glsentrydescplural{\glslabel}}{\glslabel}}%
      {\glsymbolpluralaccessdisplay
        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
    }%
  }%
}
\glscapscase
{%
  \ifglused\glslabel
  {%
    #2{\glstextaccessdisplay
      {\glsentrytext{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glsymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
    }%
  }%
  #1{\glsfirstaccessdisplay
    {\glsentryfirst{\glslabel}}{\glslabel}}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}
\ifglused\glslabel

```

```

#2{\glstextaccessdisplay
  {\Glsentrytext{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\glsentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\glsentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\Glsentryfirst{\glslabel}}{\glslabel}}%
{\glsdescriptionaccessdisplay
  {\glsentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay
  {\glsentrysymbol{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
  \MakeUppercase{%
    #2{\glstextaccessdisplay
      {\glsentrytext{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glssymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
    }%
    {%
      \MakeUppercase{%
        #1{\glsfirstaccessdisplay
          {\glsentryfirst{\glslabel}}{\glslabel}}%
          {\glsdescriptionaccessdisplay
            {\glsentrydesc{\glslabel}}{\glslabel}}%
          {\glssymbolaccessdisplay
            {\glsentrysymbol{\glslabel}}{\glslabel}}%
          {\glsinsert}}%
        }%
      }%
    }%
  }%
}
\ifglsused{\glslabel}%
{%
#2{\glscustomtext}%
{\glsdescriptionaccessdisplay
  {\glsentrydesc{\glslabel}}{\glslabel}}%
{\glssymbolaccessdisplay

```

```

        {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}%
}%
{
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}%
}%
}%
}
\renewcommand*{\glsgenentryfmt}{%
\ifdefempty\glscustomtext
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \ifglsused\glslabel
            {%
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
            {%
                \glsfirstpluralaccessdisplay
                {\glsentryfirstplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
        }%
    }%
    {%
        \ifglsused\glslabel
        {%
            \glspluralaccessdisplay
            {\Glsentryplural{\glslabel}}{\glslabel}%
            \glsinsert
        }%
        {%
            \glsfirstpluralaccessdisplay
            {\Glsentryfirstplural{\glslabel}}{\glslabel}%
            \glsinsert
        }%
    }%
    {%
        \ifglsused\glslabel
        {%
            \glspluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%

```



```

    {%
      \glscustomtext\glsinsert
    }%
  }
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglsused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
            {\glsshortpluralaccessdisplay
              {\glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
      }%
      {%
        \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
        \glsinsert
      }%
      {%
        \mfirstucMakeUppercase
        {\acronymfont
          {\glsshortpluralaccessdisplay
            {\glsentryshortpl{\glslabel}}{\glslabel}}%
        \glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \acronymfont
        {\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
  }%
  {%
    \acronymfont
      {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
    \glsinsert
  }%
  {%
    \mfirstucMakeUppercase
    {\acronymfont{%
      \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
    \glsinsert}%
  }%
}

```

```

    }%
  }%
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \genplacrfullformat{\glslabel}{\glsinsert}%
      }%
      {%
        \Genplacrfullformat{\glslabel}{\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\genplacrfullformat{\glslabel}{\glsinsert}}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}
}
}
}
\glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%

```

```

\glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
(\glsshortpluralaccessdisplay
  {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay
        {\MakeUppercase{\glsentryshort{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gls@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glsifcaps\@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\glsifcaps\@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\glsifcaps\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glsentrytext{#2}}{#2}}#3%
}
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}}#3%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glstextaccessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
  \mfirstucMakeUppercase{#3}}%
}

```

```

}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glentryfirst{#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{\glentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glentryplural{#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{\glentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glentryfirstplural{#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{\glentryfirstplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glentryname{#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{\glentryname{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

```

```

}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}#{2}%
  {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}%
  {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}#{2}%
  {\glsymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsymbolpluralaccessdisplay{\glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}#{2}%
  {\glsymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbolplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}#{2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}#{2}%
  {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
}

```

```

\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}%
  }
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}
\def\@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*\glossentrydesc[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}}%
    {\glsymbolaccessdisplay{#5}{#2}}{#6}%
  }
\renewacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
  (\glsshortaccessdisplay
    {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
  (\glsshortaccessdisplay
    {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space

```

```

(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%

```

```

\glscapscase
{%
  \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
  \glsinsert
}%
{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
  \glsinsert
}%
{%
  \glslongpluralaccessdisplay
  {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
  \mfirstucMakeUppercase{\glsinsert}%
}%
}%
{%
  \glscapscase
  {%
    \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
  }%
  {%
    \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
  }%
  {%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase
     {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
\glsgenentryfmt
}%
{\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glslongaccessdisplay

```

```

        {\mfirstucMakeUppercase{\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}%
{%
\ifglsashaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2}##3\space
(\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}{##2}##3\space
(\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsshortaccessdisplay
{\mfirstucMakeUppercase
{\acronymfont{\glsentryshort{##2}}{##2}##3\space
(\glslongaccessdisplay{\glsentrylong{##2}}{##2})}}%
\renewcommand*{\acrfullplfmt}[3]{%

```

```

\glslink[##1]{##2}{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\Glsentryshortpl{##2}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glsentryshortpl{##2}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2})}}}}%
\renewcommand*\glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##1}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}{##1}\space
    (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})}%
}
\renewacronymstyle{footnote-sc}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*\acrpluralsuffix{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{
  \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1)}}%
}
\newcommand*{\glsdefaultshortaccess}[2]{#1}
\renewcommand*{\newacronymhook}{%
\protected@edef\@gls@keylist{%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
\the\glskeylisttok}%
\expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*{\DefaultNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
}%
}

```

```

type=\acronymtype,%
name={\the\glsshorttok},%
description={\the\glslongtok},%
descriptionaccess=\relax,
text={\the\glsshorttok},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\noexpand\glslongaccessdisplay
  {\the\glslongtok}{\the\glslabeltok}\space
  (\noexpand\glsshortaccessdisplay
    {\the\glsshorttok}{\the\glslabeltok})},%
plural={\the\glsshorttok\acrpluralsuffix},%
firstplural={\noexpand\glslongpluralaccessdisplay
  {\noexpand\@glo@longpl}{\the\glslabeltok}\space
  (\noexpand\glsshortpluralaccessdisplay
    {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
firstaccess=\relax,
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
\@gls@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%

```

```

type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
access={\noexpand\@glo@textaccess},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\the\glslongtok},%
symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand
      \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@gl@assign@firstpl\gl@assign@firstpl
\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@plural\@org@gl@assign@plural
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@descplural\@org@gl@assign@descplural
}
\renewcommand*\@SmallNewAcronymDef{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\gl@labeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\gl@shorttok}},%
    access={\noexpand\@glo@symbolaccess},%
    sort={\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},%
    text={\noexpand\@glo@short},%
    textaccess={\noexpand\@glo@shortaccess},%
    plural={\noexpand\@glo@shortpl},%
    first={\the\gl@longtok},%
    firstaccess=\relax,
    firstplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
    description={\noexpand\@glo@first},%
    descriptionplural={\noexpand\@glo@firstplural},%
    symbol={\the\gl@shorttok},%
    symbolaccess={\the\gl@longtok},%
    symbolplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
    \the\gl@keylisttok
  }%
}%
\let\@org@gl@assign@firstpl\gl@assign@firstpl
\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\let\@org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%

```

```

\def\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*{\showglongdescaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*{\showglongdescpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*{\showglongshortaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*{\showglongshortpluralaccess}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*{\showglongaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*{\showglongpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}

```

8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
{}%

```

8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2021/11/01 v4.49 (NLCT)]
\ifglsxindy
\renewcommand*\GlsAddXdyAttribute[1]{%
\edef\xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\xdylocref{\the\toks@ ^^J}%
(markup-locref

```

```

:open \string"\string~n\string\setentrycounter
  {\noexpand\glscounter}%
  \expandafter\string\csname#1\endcsname
  \expandafter@gobble\string\{\string" ^^J
:close \string"\expandafter@gobble\string\}\string" ^^J
:attr \string"#1\string"}}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glsnumber}
\GlsAddXdyAttribute{hyper}
\GlsAddXdyAttribute{hyper}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglsxindy
  \renewcommand*\GlsAddXdyLocation[2]{%
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
        \space(#2))
    }%
    \edef\@xdyuserlocationnames{%
      \@xdyuserlocationnames^^J\space\space\space
      \string"#1\string"}%
  }
\fi
\renewcommand{\do@wrglossary}[1]{%
\ifglsxindy
  \expandafter\glo@check@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax

```

```

\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
\protected@edef\@glslocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@glslocref
\glossary[\csname glo@#1@type\endcsname]{%
(indexentry :tkey (\csname glo@#1@index\endcsname)
:locref \string"\@glslocref\string" %
:attr \string"\@glo@suffix\string" \@glo@range
)
}%
\else
\@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
\glossary[\csname glo@#1@type\endcsname]{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar\@glo@numfmt}{\theglsentrycounter}}%
\fi
}
\def\@set@glo@numformat#1#2#3{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[] {#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\ifglxindy
\def\writeist{%
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package in compatible-2.07 mode}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%

```

```

\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-lowercase\string")
    :min-range-length \@glsminrange)}%
\else
  \write\glswrite{(define-location-class
    \string"roman-page-numbers\string"^^J\space\space\space
    (:sep "\@gls@roman")
    :min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
  \string"Roman-page-numbers\string"^^J\space\space\space
  (\string"roman-numbers-uppercase\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-page-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"alpha-page-numbers\string"^^J\space\space\space
  (\string"alpha\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
  :sep \string"\@glsAlphacompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string")
  :min-range-length \@glsminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%

```

```

\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glseeformat\string"
: close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble\string~n\string\begin
{theglossary}\string\glossaryheader\string~n\string" ^^J\space
\space\space:close \string"\expandafter\@gobble
\string~%\string~n\string
\end{theglossary}\string\glossarypostamble
\string~n\string" ^^J\space\space\space
: tree})}%
\write\glswrite{(markup-letter-group-list
: sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
: open \string"\string\relax \string\glsresetentrylist
\string~n\string")}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
: sep \string", \string"
: close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
: sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
: sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
: close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
: close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
: sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{^^J; letter group headings^^J}%

```

```

\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string

```

```

    \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*\noist{\let\writeist\relax}

```

8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
  }%
}

```

```

\glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
\ifx\glo@desc\@no@post@desc
  \glsinlineemptydescformat{##4}{##5}%
\else
  \ifstrempy{##3}%
    {\glsinlineemptydescformat{##4}{##5}}%
    {\glsinlinedescformat{##3}{##4}{##5}}%
  \fi
\ifglshaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{##3}%
  \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
}
\compatglossarystyle{list}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
      ##3\glspostdescription\space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]%
      \mbox{}\par\nobreak\@afterheading
      ##3\glspostdescription\space ##5}%
  \renewcommand{\glossarysubentryfield}[6]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%

```

```

\csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
\csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
\renewcommand*{\glossaryentryfield}[5]{%
\item[]\makebox[\glslistdottedwidth][l]{%
\glsentryitem{##1}\glstarget{##1}{##2}%
\unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
\renewcommand*{\glossarysubentryfield}[6]{%
\item[]\makebox[\glslistdottedwidth][l]{%
\glssubentryitem{##2}%
\glstarget{##2}{##3}%
\unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
}%
\compatglossarystyle{sublistdotted}{%
\csuse{@glscompstyle@listdotted}%
\renewcommand*{\glossaryentryfield}[5]{%
\item[\glsentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{longborder}{%
\csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
\csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
\csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%

```

```

\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
\csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
\csuse{@glscompstyle@longragged}%
}%

```

```

\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glsentryitem{##1}\textbf{\glstarget{##1}{##2}}%
    \ifx\relax##4\relax
    \else
    \space{##4}%
    \fi
    \space ##3\glspostdescription \space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \ifcase##1\relax
    % level 0
    \item
    \or

```

```

        % level 1
        \subitem
        \glssubentryitem{##2}%
    \else
        % all other levels
        \subsubitem
    \fi
    \textbf{\glstarget{##2}{##3}}%
    \ifx\relax##5\relax
    \else
        \space{##5}%
    \fi
    \space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
\csuse{@glscmpstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
\csuse{@glscmpstyle@index}%
}%
\compatglossarystyle{tree}{%
\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
\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
  \glentryitem{##1}\textbf{\glstarget{##1}{##2}}%
  \ifx\relax##4\relax
  \else
    \space{##4}%
  \fi
  \space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  ##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
  \csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
  \csuse{@glscompstyle@treenoname}%
}%
\compatglossarystyle{almtree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \ifnum\@gls@prevlevel=0\relax
    \else
      \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
      \hangindent\glstreeindent
      \parindent\glstreeindent
    \fi
    \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
      \glentryitem{##1}\textbf{\glstarget{##1}{##2}}}}%
    \ifx\relax##4\relax
    \else
      (##4)\space
    \fi
    ##3\glspostdescription \space ##5\par
    \def\@gls@prevlevel{0}%
  }%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
      \settowidth{\gls@tmplen}{\textbf{%

```

```

        \csname @glswidestname\romannumeral##1\endcsname\space}}}%
\ifnum@gls@prevlevel<##1\relax
  \setlength\glstreeindent\gls@tmplen
  \addtolength\glstreeindent\parindent
  \parindent\glstreeindent
\else
  \@ifundefined{glswidestname\romannumeral@gls@prevlevel}{%
    \settowidth{\glstreeindent}{\textbf{%
      \glswidestname\space}}}%
    \settowidth{\glstreeindent}{\textbf{%
      \csname @glswidestname\romannumeral@gls@prevlevel
        \endcsname\space}}}%
    \addtolength\parindent{-\glstreeindent}%
    \setlength\glstreeindent\parindent
  \fi
\fi
\hangindent\glstreeindent
\makebox[0pt][r]{\makebox[\gls@tmplen][l]{%
  \textbf{\glstarget{##2}{##3}}}}%
\ifx##5\relax\relax
\else
  (##5)\space
\fi
##4\glspostdescription\space ##6\par
\def@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreegroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
  \csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
  \csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
  \csuse{@glscompstyle@tree}%
}

```

```

}%
\compatglossarystyle{mcoltreenoname}{%
  \csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
  \csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
  \csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
  \csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
  \csuse{@glscompstyle@almtree}%
}%

\compatglossarystyle{superragged}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glstarget{##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]{%
    \glstarget{##1}{\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%

```

```

\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscompstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \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}%
}%
}%

```

8.23 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*\glentryprefixfirst}[1]{\csuse{glo@glstetoklabel{#1}@prefixfirst}}
\newcommand*\glentryprefixfirstplural}[1]{%
  \csuse{glo@glstetoklabel{#1}@prefixfirstplural}}
\newcommand*\glentryprefix}[1]{\csuse{glo@glstetoklabel{#1}@prefix}}
\newcommand*\glentryprefixplural}[1]{\csuse{glo@glstetoklabel{#1}@prefixplural}}
\newrobustcmd*\Glentryprefixfirst}[1]{%
  \protected@edef\@glo@text{\csname glo@glstetoklabel{#1}@prefixfirst\endcsname}%
  \xmakefirstuc\@glo@text
}
\newrobustcmd*\Glentryprefixfirstplural}[1]{%
  \protected@edef\@glo@text{\csname glo@glstetoklabel{#1}@prefixfirstplural\endcsname}%
  \xmakefirstuc\@glo@text
}
\newrobustcmd*\Glentryprefix}[1]{%
  \protected@edef\@glo@text{\csname glo@glstetoklabel{#1}@prefix\endcsname}%
  \xmakefirstuc\@glo@text
}
\newrobustcmd*\Glentryprefixplural}[1]{%
  \protected@edef\@glo@text{\csname glo@glstetoklabel{#1}@prefixplural\endcsname}%
  \xmakefirstuc\@glo@text
}
\newcommand*\ifglshasprefix}[3]{%
  \ifcsemtyp{glo@glstetoklabel{#1}@prefix}%
  {#3}%
  {#2}%
}
\newcommand*\ifglshasprefixplural}[3]{%
  \ifcsemtyp{glo@glstetoklabel{#1}@prefixplural}%
  {#3}%
  {#2}%
}
\newcommand*\ifglshasprefixfirst}[3]{%
  \ifcsemtyp{glo@glstetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*\ifglshasprefixfirstplural}[3]{%
  \ifcsemtyp{glo@glstetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glstr@record
  {\@glstr@record{#1}{#2}{glslink}}%
  {}%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
\newcommand*\@pgls}[2][1]{%
  \new@ifnextchar[%

```

```

    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2} []}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglsashasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglsashasprefixfirst{#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}%
    {%
      \ifglsashasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglsashasprefixfirstplural{#2}%
      {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2} [#3]%
  }%
}
\newrobustcmd{\PglS}{\@gls@hyp@opt\@PglS}
\newcommand*{\@PglS}[2] []{%
  \new@ifnextchar [%
  {\@PglS@{#1}{#2}}%
  {\@PglS@{#1}{#2} []}%
}
\def\@PglS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%

```

```

\ifglshasprefix{#2}%
{%
  \Glentryprefix{#2}%
  \glsprefixsep
  \@gls@{#1}{#2}[#3]%
}%
{\@Gls@{#1}{#2}[#3]}%
}%
{%
\ifglshasprefixfirst{#2}%
{%
  \Glentryprefixfirst{#2}%
  \glsprefixsep
  \@gls@{#1}{#2}[#3]%
}%
{\@Gls@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\Pglsp1}{\@gls@hyp@opt\@Pglsp1}
\newcommand*{\@Pglsp1}[2][{}]{%
  \new@ifnextchar[{}]{\@Pglsp1@{#1}{#2}}%
  {\@Pglsp1@{#1}{#2}[{}]}%
}
\def\@Pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirstplural{#2}%
      {%
        \Glentryprefixfirstplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}%
}

```

```

\newrobustcmd{\PGLS}{\@gls@hyp@opt\PGLS}
\newcommand*{\@PGLS}[2][\%
  \new@ifnextchar[\%
    {\@PGLS@{#1}{#2}}{\%
    {\@PGLS@{#1}{#2}[]}\%
  }
\def\@PGLS@#1#2[#3]{\%
  \@glsprefix@record@hook{#1}{#2}\%
  \glsdoifexists{#2}\%
  {\%
    \ifglsused{#2}\%
    {\%
      \ifglshasprefix{#2}\%
      {\mfirstucMakeUppercase{\glsentryprefix{#2}\glsprefixsep}}{\}%
    }%
    {\%
      \ifglshasprefixfirst{#2}\%
      {\mfirstucMakeUppercase{\glsentryprefixfirst{#2}\glsprefixsep}}{\}%
    }%
  }%
  \@GLS@{#1}{#2}[#3]\%
}%
}
\newrobustcmd{\PGLSp1}{\@gls@hyp@opt\PGLSp1}
\newcommand*{\@PGLSp1}[2][\%
  \new@ifnextchar[\%
    {\@PGLSp1@{#1}{#2}}{\%
    {\@PGLSp1@{#1}{#2}[]}\%
  }
\def\@PGLSp1@#1#2[#3]{\%
  \@glsprefix@record@hook{#1}{#2}\%
  \glsdoifexists{#2}\%
  {\%
    \ifglsused{#2}\%
    {\%
      \ifglshasprefixplural{#2}\%
      {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{\}%
    }%
    {\%
      \ifglshasprefixfirstplural{#2}\%
      {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{\}%
    }%
  }%
  \@GLSp1@{#1}{#2}[#3]\%
}%
}

```

8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (NLCT)]
```

```

\newcommand*\glsnavhyperlink}[3][\@glo@type]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*\glsnavhyperlinkname}[2]{glsn:#1@#2}
\newcommand*\glsnavhypertarget}[3][\@glo@type]{%
  \@glsnavhypertarget{#1}{#2}{#3}}
}
\newcommand*\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}\@string\@gls@hypergroup{#1}{#2}}%
  \@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@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@hypergroup@list@#1\endcsname
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
      \@gls@tmp,#2}}%
  }%
}
\newcommand*\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{\@gls@hypergroup@list@\@glo@type}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \expandafter\let\expandafter\@gls@list
      \csname @gls@hypergroup@list@\@glo@type\endcsname
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}}%
    \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}}%
    \let\@gls@between\gls@hypernavsep

```

```

    }%
  }
  \newcommand*{\glshypernavsep}{\space\textbar\space}
  \newcommand*{\glssymbolnav}{%
    \glshyperlink{glssymbols}{\glsgrouptitle{glssymbols}}%
    \glshypernavsep
    \glshyperlink{glsnumbers}{\glsgrouptitle{glsnumbers}}%
    \glshypernavsep
  }

```

8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2021/11/01 v4.49 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{}%
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{}%
  }%
  {\glspostinline}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \glsinlinedopostchild
  \gls@inlinesep
  \glsentryitem{##1}%
  \glsinlinenameformat{##1}%
  \glossentryname{##1}%
}%
\ifglsdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
\ifglsdesc{##1}%
{\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
{\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\ifglsdesc{##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
}%
\ifglsnogroupskip
  \renewcommand*\glsgroupskip{}%
\else
  \renewcommand*\glsgroupskip{ & \tabularnewline}%
\fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|p{\glsdescwidth}|}{\end{longtable}}%
    \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%
  \renewcommand*\glossaryheader{%
    \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
  }
\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*\glossaryheader{%
    \hline\bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{long3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}p{\glspagelistwidth}}%
    {\end{longtable}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
  }

```

```

        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
\ifglsnogroupskip
    \renewcommand*\glsgroupskip}{}%
\else
    \renewcommand*\glsgroupskip}{ & & \tabularnewline}%
\fi
}
\newglossarystyle{long3colborder}{%
    \setglossarystyle{long3col}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
        {\end{longtable}}}%
    \renewcommand*\glossaryheader{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
    \setglossarystyle{long3col}%
    \renewcommand*\glossaryheader{%
        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
    \setglossarystyle{long3colborder}%
    \renewcommand*\glossaryheader{%
        \hline
        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline\endhead
        \hline\endfoot}%
}
\newglossarystyle{long4col}{%
    \renewenvironment{theglossary}%
        {\begin{longtable}{|l|l|l|l|}}%
        {\end{longtable}}}%
    \renewcommand*\glossaryheader}{}%
    \renewcommand*\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glstarget{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} &
        \glossentrysymbol{##1} &
        ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        \glossentrysymbol{##2} & ##3\tabularnewline
    }%
\ifglsnogroupskip
    \renewcommand*\glsgroupskip}{}%

```

```

\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%

```

```

    {\begin{longtable}{|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
    {\end{longtable}}%
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
}

```

```

\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\lp{\glstdescwidth}\lp{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glstdescwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glstdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\l>{\raggedright}p{\glstdescwidth}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*{\glstoreLToutput}{\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\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]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1}\glspostdescription\space ##2%
}

```

```

        \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}%
        \glspostdescription\space ##3%
        \tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*\glsgroupskip{}%
    \else
        \renewcommand*\glsgroupskip{ & \tabularnewline}%
    \fi
}
\newglossarystyle{longraggedborder}{%
    \setglossarystyle{longragged}%
    \renewenvironment{theglossary}{%
        \begin{longtable}{|l|>{\raggedright}p{\glstdescwidth}|}%
        {\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{\glstdescwidth}%
                >{\raggedright}p{\glspagelistwidth}}}%
            {\end{longtable}}%
        \renewcommand*\glossaryheader{}%
        \renewcommand*\glsgroupheading[1]{}%
        \renewcommand{\glossentry}[2]{%
            \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
            \glossentrydesc{##1} & ##2\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
            &
            \glssubentryitem{##2}%

```

```

        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
\ifglsgroupskip
    \renewcommand*\glsgroupskip{}%
\else
    \renewcommand*\glsgroupskip{ & & \tabularnewline}%
\fi
}
\newglossarystyle{longragged3colborder}{%
    \setglossarystyle{longragged3col}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{|l|>{\raggedright}p{\glstdescwidth}|%
            >{\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{\glstdescwidth}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}%
}
}

```

8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.49]
\providecommand{\indexspace}{%
    \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
    \setglossarystyle{index}%
}

```

```

\renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
    \let\item\glstreeitem
    \let\subitem\glstreesubitem
    \let\subsubitem\glstreesubsubitem
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem}%
    {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
}
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%

```

```

\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*\glossaryheader{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreesspannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
\renewcommand*\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsncols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*\glossaryheader{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

```

}
\newglossarystyle{mcoltreononamespannav}{%
  \setglossarystyle{treononame}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtree}{%
  \setglossarystyle{almtree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalmtreegroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalmtreehypergroup}{%
  \setglossarystyle{mcolalmtree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalmtreespannav}{%

```

```

\setglossarystyle{almtree}%
\renewenvironment{theglossary}%
{%
  \begin{multicols}{\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{\glsnavigationtarget{##1}{\glssetgrouptitle{##1}}}\par
  \indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
  {\tablehead{} \tabletail{}%
  \begin{supertabular}{lp{glsdescwidth}}%
  {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%

```

```

\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}{|l|p{\glsdescwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
     \bfseries\descriptionname\tabularnewline}%
     \tabletail{}%
     \begin{supertabular}{|lp{\glsdescwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
     \bfseries \descriptionname\tabularnewline\hline}%
     \tabletail{\hline}
     \begin{supertabular}{|l|p{\glsdescwidth}|}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
     \begin{supertabular}{|lp{\glsdescwidth}p{\glspagelistwidth}|}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%

```

```

\fi
}
\newglossarystyle{super3colborder}{%
\setglossarystyle{super3col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
{\end{supertabular}}}%
}
\newglossarystyle{super3colheader}{%
\setglossarystyle{super3col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline}\tabletail{}}%
\begin{supertabular}{|p{\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]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
      \tabletail{}}%
    {\begin{supertabular}{|l|l|l|l|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
      \begin{supertabular}{|l|l|l|l|}}%
    {\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}{|p{\glsdescwidth}|p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{|p{\glsdescwidth}|p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
      \begin{supertabular}%
        |l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}}%
}

```

```

        {\end{supertabular}}}%
    }
\newglossarystyle{altsuper4colheaderborder}{%
  \setglossarystyle{super4colheaderborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {1|p{\glsdescwidth}|1|p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
  }

```

8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
      \begin{supertabular}{1>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%}
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
}

```

```

\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superraggedheader}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname & \bfseries \descriptionname
     \tabularnewline}%
     \tabletail{}}%
     \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superraggedheaderborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
     \bfseries \descriptionname\tabularnewline\hline}%
     \tabletail{\hline}
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superragged3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
     \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
     >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
}

```

```

\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
  \setglossarystyle{superragged3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superragged3colheader}{%
  \setglossarystyle{superragged3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
     \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{superragged3colheaderborder}{%
  \setglossarystyle{superragged3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
     \tabletail{\hline}%
     \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{} \tabletail{}%
     \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%

```

```

        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsnogroupskip
        \renewcommand*\glsgroupskip{}%
    \else
        \renewcommand*\glsgroupskip}{& & \tabularnewline}%
    \fi
}
\newglossarystyle{altsuperragged4colheader}{%
\setglossarystyle{altsuperragged4col}%
\renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{l>{\raggedright}p{\glstdescwidth}l%
>{\raggedright}p{\glspagelistwidth}}}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4colborder}{%
\setglossarystyle{altsuper4col}%
\renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
    {l|>{\raggedright}p{\glstdescwidth}l|l%
>{\raggedright}p{\glspagelistwidth}l}}%
\end{supertabular}}%
}
\newglossarystyle{altsuperragged4colheaderborder}{%
\setglossarystyle{altsuperragged4col}%
\renewenvironment{theglossary}%
    {\tablehead{\hline
\bfseries\entryname &
\bfseries\descriptionname &
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
    {l|>{\raggedright}p{\glstdescwidth}l|l%
>{\raggedright}p{\glspagelistwidth}l}}%
\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}{\ifglsgnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
  \setglossarystyle{index}%

```

```

\renewcommand*\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
  \indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
  \setglossarystyle{index}%
  \renewcommand*\glossaryheader{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
    \indexspace}%
}
\newglossarystyle{tree}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
  }%
  \renewcommand*\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*\glossaryheader{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*\glsgroupheading}[1]{%

```

```

\par\noindent
\glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}\par
  \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstarget{##2}{\strut}%
    \glossentrydesc{##2}\glspostdescription\space##3\par
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
  \noindent\glstreegroupheaderfmt
  {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}\par
    \indexspace}%
  }
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax

```

```

\gls@tmplen=0pt\relax
\forallglossaries[#1]{\@gls@type}%
{%
  \forallglsentries[\@gls@type]{\@glo@label}%
  {%
    \ifglsashasparent{\@glo@label}%
    {}%
    {%
      \settowidth{\dimen@}%
      {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
      \gls@tmplen=\dimen@
      \letcs{\@glswidestname}{glo@\glsdetoklabel{\@glo@label}@name}%
      \fi
    }%
  }%
}%
}
\newcommand*\glssetwidest[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
    #2}%
}
\newcommand*\@glswidestname{}
\newcommand*\glstreenamebox[2]{%
  \makebox[#1][l]{#2}%
}
\newglossarystyle{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}{%
        \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
      \ifgls hassymbol{##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}%

```

```

\hangindentOpt\relax
\parindentOpt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.35 Rollback v4.52 (glossaries-2022-11-03.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareCurrentRelease{v4.52}{2022-11-03}
\ProvidesPackage{glossaries}[2022/11/03 v4.52 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\newcommand{\glsentencecase}[1]{\makefirstuc{#1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{\text_lowercase:n{#1}}
\newcommand{\glsuppercase}[1]{\text_uppercase:n{#1}}
\ifdef\MFUsentencecase
{%
\newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
\newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
\newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
}
{
\RequirePackage{textcase}
\providecommand{\MFUsentencecase}[1]{\text_titlecase_first:n{#1}}
\providecommand{\glsmfuexcl}[1]{
\tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
{
\tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
}
}
\renewcommand*\mfirstucMakeUppercase{\MakeTextUppercase}
\newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
\newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
{%
\@gls@docloadedtrue
}%
}%

```

```

\@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
}
\if@gls@docloaded
\let\glsorg@theglossary\theglossary
\let\glsorg@endtheglossary\endtheglossary
\let\glsorg@PrintChanges\PrintChanges
\renewcommand{\PrintChanges}{%
\begingroup
\let\theglossary\glsorg@theglossary
\let\endtheglossary\glsorg@endtheglossary
\glsorg@PrintChanges
\endgroup
}
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\@gls@debug@val\@gls@debug@nr]%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\@gls@debug@nr\relax
% debug=true
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@gls@showtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@gls@showtarget\@gobble
\let\@gls@showaccsupp\@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{\@gls@showtarget}{\@gls@showtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}

```

```

}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
  \ifmmode
    \glsshowtargetinner{#1}%
  \else
    \ifinner
      \glsshowtargetinner{#1}%
    \else
      \glsshowtargetouter{#1}%
    \fi
  \fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
  \ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{%
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining

```

```

any entries that have a ‘\gls@xr@key’ key. It may
be that the ‘see’ key has been written to the .glsdefs
file from the previous run, in which case you need to
move your definitions
to the preamble if you don’t want to use
\string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}%
[\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@esclocationfalse
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}}
\@gls@declareoption{nonolong}{\renewcommand*{\@gls@loadlong}{}}
\IfFileExists{supertabular.sty}{%
    \newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}}}%
    \newcommand*{\@gls@loadsuper}{}}
\@ifpackageloaded{xtab}{\renewcommand*{\@gls@loadsuper}{}}{}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}}
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}}
\@gls@declareoption{nolist}{%
    \renewcommand*{\@gls@loadlist}{%
        \ifdefstring{\@glossary@default@style}{list}%
        {\let\@glossary@default@style\relax}%
        }%
}
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
\@gls@declareoption{nostyles}{%

```

```

\renewcommand*{\@gls@loadlong}{}%
\renewcommand*{\@gls@loadsuper}{}%
\renewcommand*{\@gls@loadlist}{}%
\renewcommand*{\@gls@loadtree}{}%
\let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode`. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
    \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  {}%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glsentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glsentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand*{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglsentrycounter
    \newcounter{glossarysubentry}[glossaryentry]%
    \else

```

```

        \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
    \fi
}%
}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{%
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglsexindy xindy\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
\renewcommand*{\@glo@default@sorttype}{#1}%
\csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
\glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{%
\newcommand*{\@gls@setupsort@standard}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\ifx\@glo@sort\@glsdefaultsort
\let\@glo@sort\@glo@name
\fi
\let\glsdosanitizesort\@gls@sanitizesort
\glsprestandardsort{\@glo@sort}{##1}{##2}%
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%

```

```

    {\@gls@defsortcount{##1}}%
    {}%
    \expandafter\global\expandafter
    \advance\csname glossary@##1@sortcount\endcsname by 1\relax
    \expandafter\protected\xdef\csname glo@##2@sort\endcsname{%
      \expandafter\glssortnumberfmt
      {\csname glossary@##1@sortcount\endcsname}}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@use}{%
  \let\do@glo@storeentry\@gobble
  \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
  }%
  \def\@gls@defsort##1##2{%
    \expandafter\gdef\csname glo@##2@sort\endcsname}%
  }%
  \def\@gls@setsort##1{%
    \protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
    \ifx\@glo@parent\@empty
    \else
      \expandafter\@gls@setsort\expandafter{\@glo@parent}%
    \fi
    \protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
    \protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
    \ifx\@gls@tmp\@empty
      \expandafter\global\expandafter
      \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
      \expandafter\protected\xdef\csname glo@##1@sort\endcsname{%
        \expandafter\glssortnumberfmt
        {\csname glossary@\@glo@type @sortcount\endcsname}}%
      \@glo@storeentry{##1}%
    \fi
  }%
  \let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
  }%
  \def\@gls@setsort##1{%
  \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
  {Option sort=none not allowed with \string##1}%
  {(Use sort=def instead)}}%

```

```

}
\newcommand*{\@gls@setupsort@clear}{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
  \def\@gls@setsort##1{%
  \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
    {Option sort=clear not allowed with \string##1}%
    {(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
  \ifgls@docloaded
    \newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
  \else
    \newglossary{main}{gls}{glo}{\glossaryname}%
  \fi
  \newcommand*{\gls@tr@set@main@toctitle}{%
    \translatelet{\glossarytoctitle}{Glossary}%
  }%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
  \let\glsdefaulttype\relax
  \renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
  \ifglsacronym
    \renewcommand{\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*{\acronymtype}{acronym}%
      \newcommand*{\gls@tr@set@acronym@toctitle}{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%
  \ifglsacronym
    \providecommand*{\printacronyms}[1][1]{%
      \printglossary[type=\acronymtype,#1]%
    }%
  \fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
}

```

```

\def\@gls@do@acronymsdef{%
  \DeclareAcronymList{acronym}%
  \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
  \renewcommand*\@acronymtype{acronym}%
  \newcommand*\gls@tr@set@acronym@to@title{%
    \translatelet{\glossaryto@title}{Acronyms}%
  }%
}%
}
\newcommand*\@gls@acronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@gls@acronymlists\@empty
    \protected@xdef\@gls@acronymlists{#1}%
  \else
    \protected@xdef\@gls@acronymlists{\@gls@acronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@gls@acronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms[4]{%
  \def\@gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*\gls@checkisacronymlist[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\SetAcronymLists[1]{%
  \renewcommand*\@gls@acronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand*\gls@counter{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*\gls@counter{#1}%
}
}

```

```

\newcommand*\@gls@nohyperlist{}
\newcommand*\GlsDeclareNoHyperList[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*\GlossariesWarning[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*\GlossariesWarningNoLine[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand*\glosortentrieswarning{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*\GlossariesWarning[1]{}%
    \renewcommand*\GlossariesWarningNoLine[1]{}%
    \renewcommand*\glosortentrieswarning{}%
    \renewcommand*\@gls@missinglang@warn[2]{}%
  \fi
}
\newcommand*\@gls@missinglang@warn[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*\@gls@missinglang@warn[2]{}%
}
\newcommand*\@gls@warnonglossdefined{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheglossdefined{%
  \GlossariesWarning{Overriding 'theglossary' environment}%
}
\@gls@declareoption{noredefwarn}{%

```

```

\renewcommand*{\@gls@warnonglossdefined}{}%
\renewcommand*{\@gls@warnontheglossdefined}{}%
}
\newcommand*{\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*{\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
\ifgls@sanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort
\fi
}
\newcommand*\@gls@sanitizesort{%
\@onelevel@sanitize\@glo@sort
}
\newcommand*\@gls@nosanitizesort{}
\newcommand*\@gls@noidx@sanitizesort{%
\ifdefvoid\@glo@sort
{}%
{%
\expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
\def\@glo@sort{#1#2}%
\@onelevel@sanitize\@glo@sort
}
\newcommand*\@gls@noidx@nosanitizesort{%
\ifdefvoid\@glo@sort
{}%
{%
\expandafter\@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
}

```

```

\def\@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x@\@secondoftwo
  \let\@tabacckludge@\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let'\@firstofone
  \let'\@firstofone
  \let^\@firstofone
  \let"@@firstofone
  \letu\@firstofone
  \lett\@firstofone
  \letd\@firstofone
  \letr\@firstofone
  \let=\@firstofone
  \let.\@firstofone
  \let~\@firstofone
  \letv\@firstofone
  \letH\@firstofone
  \letc\@firstofone
  \letb\@firstofone
  \leta\@secondoftwo
  \def\AE{AE}%
  \def\ae{ae}%
  \def\OE{OE}%
  \def\oe{oe}%
  \def\AA{AA}%
  \def\aa{aa}%
  \def\L{L}%
  \def\l{l}%
  \def\O{O}%
  \def\o{o}%
  \def\SS{SS}%
  \def\ss{ss}%
  \def\th{th}%
  \def\TH{TH}%
  \def\dh{dh}%
  \def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}

```

```

{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[gls]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifgls@sanitize@symbol
\glssetnoexpandfield{symbol}%
\glssetnoexpandfield{symbolplural}%
\else
\glssetexpandfield{symbol}%
\glssetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
\ifgls@sanitizesort
\glssetnoexpandfield{sortvalue}%
\renewcommand*{\@gls@noidx@setsanitizesort}{%
\gls@sanitizesorttrue
\glssetnoexpandfield{sortvalue}%
}%
\else
\glssetexpandfield{sortvalue}%
\renewcommand*{\@gls@noidx@setsanitizesort}{%
\gls@sanitizesortfalse
\glssetexpandfield{sortvalue}%
}%
}

```

```

    \fi
  }
  \glssanitizesorttrue
  \glsssetnoexpandfield{sortvalue}%
  \newcommand*\@gls@noidx@setsanitizesort}{%
    \glssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
  }
  \define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glssanitizesort}{#1}%
    \ifglssanitizesort
      \glsssetnoexpandfield{sortvalue}%
    \else
      \glsssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
      deprecated. Use sanitizesort instead}%
  }
  \define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
      {%
        \GlossariesWarning{sanitize package option deprecated}%
        \glsssetexpandfield{name}%
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
        \glsssetexpandfield{desc}%
        \glsssetexpandfield{descplural}%
      }%
    }{%
      \setkeys[gls]{sanitize}{#1}%
    }%
  }
  \newif\ifglstranslate
  \newcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \@ifpackageloaded{babel}%
      {%
        \IfFileExists{translator.sty}%
        {%
          \RequirePackage{translator}%
          \let\glsifusetranslator\@firstoftwo
        }%
      }%
    }%
  }%

```

```

}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%
    \ifcase\gls@translate@nr\relax
      \glstranslatetrue
      \renewcommand*\@gls@usetranslator{%
        \@ifpackageloaded{polyglossia}%
        {%
          \let\glsifusetranslator\@secondoftwo
        }%
        {%
          \@ifpackageloaded{babel}%
          {%
            \IfFileExists{translator.sty}%
            {%
              \RequirePackage{translator}%
              \let\glsifusetranslator\@firstoftwo
            }%
            {}%
          }%
          {}%
        }%
      }%
    \or
      \glstranslatefalse
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \or
      \glstranslatetrue
      \let\@gls@usetranslator\relax
      \let\glsifusetranslator\@secondoftwo
    \fi
  }
  \glstranslatefalse
  \let\glsifusetranslator\@secondoftwo
  \@ifpackageloaded{translator}%
  {%
    \glstranslatetrue

```

```

\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\endfortrue
}%
{}%
}
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%
\renewcommand{\@gls@requested@languages}{#1}%
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{%
\glsindexonlyfirstfalse
}
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{%
\glshyperfirsttrue
}
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{%
\ifglsacrfootnote
\@gls@removedoption{footnote}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{%
\ifglsacrdescription
\@gls@removedoption{description}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{%
\ifglsacrsmallcaps
\@gls@removedoption{smallcaps}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\ifglsacrsmaller
\@gls@removedoption{smaller}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\ifglsacrdua
\@gls@removedoption{dua}%
\fi
}
}

```

```

\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*{\glsorder}{word}
\newcommand*{@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{%
  \def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}[glsnumbers][true]{}
\gls@xindy@glsnumberstrue
\def\xdy@main@language{\languagename}%
\newcommand*{gls@xdy@set@lang}[1]{}
\ifcsundef{xdy@#1@language}%
  {%
    \ifcsundef{xdy@glsdefaulttype @language}
    {\let@gls@langmod\xdy@main@language}%
    {\letcs@gls@langmod{xdy@glsdefaulttype @language}}%
  }%
  {\letcs@gls@langmod{xdy@#1@language}}%
}
\define@key[gls]{xindy}{language}{\def\xdy@main@language{#1}}
\ifdefvoid\inputencodingname
  {\def@gls@codepage{utf8}}%
  {\def@gls@codepage{\inputencodingname}}
\define@key[gls]{xindy}{codepage}{\def@gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{}
  \glsxindytrue
  \setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
  \glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
  \glsxindytrue
  \gls@xindy@glsnumbersfalse
}
\providecommand{@domakeglossaries}[1]{\if@filesw #1\fi}
\@gls@declareoption{disablemakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘disablemakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘disablemakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
  }%
}

```

```

        \renewcommand{\@domakeglossaries}[1]{%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been disabled}%
        }%
    }%
}
\@gls@declareoption{restoremakegloss}{%
    \ifdefequal\makeglossaries\@no@makeglossaries
    {%
        \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makenoidxglossaries\space has already been used)}%
    }%
    {%
        \ifdefequal\@makeglossary\@gobble
        {%
            \GlossariesWarning{Option ‘restoremakegloss’ has no effect
            (\string\makeglossaries\space has already been used)}%
        }%
        {%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been restored}%
            \let\@domakeglossaries\@firstofone
        }%
    }%
}
}
\newcommand*{\@do@write@glslabels}{%
    \AtEndDocument{\@do@write@glslabels}%
    \let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
    \newwrite\@gls@labelsfile
    \immediate\openout\@gls@labelsfile=\jobname.glslabels
    \protected@edef\@gls@fulllist{\@glo@types}%
    \ifdefempty{\@ignored@glossaries}{}%
    {\protected@eappto\@gls@fulllist{,\@ignored@glossaries}}%
    \forallglsentries[\@gls@fulllist]{\@glsentry}%
    {%
        \ifdefempty{\@glsentry}{}%
        {%
            \ifx\do@write@glslabelsname\@gobble
                \def\@gls@thisentryname{}%
            \else
                \letcs{\@gls@thisentryname}{glo@\@glsdetoklabel{\@glsentry}@name}%
                \@onelevel@sanitize\@gls@thisentryname
            \fi
            \immediate\write\@gls@labelsfile
            {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
        }%
    }%
}

```

```

\immediate\closeout\@gls@labelsfile
}
\newcommand*{\do@write@gls@labelsname}[1]{%
\begingroup
\catcode'\^^I12
\gdef\@write@gls@labelsname#1{\^^I#1}
\endgroup
\@gls@declareoption{writeglslabels}{\do@write@gls@labels
\let\do@write@gls@labelsname\@gobble}
\@gls@declareoption{writeglslabelnames}{\do@write@gls@labels
\let\do@write@gls@labelsname\@write@gls@labelsname}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\newcommand{\gls@automake@makegloss}{%}
\newcommand{\gls@automake@option}{automake=\gls@automake@val}
\newcommand*{\@gls@do@automake@err}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with \gls@automake@option}
{%
Either remove the \gls@automake@option\space setting or
add \string\makeglossaries\space to your document preamble.%
}%
}%
\define@choicekey{glossaries.sty}{automake}{%
[\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\gls@automake@nr\relax
\GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or
\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\@gls@do@automake\@gls@do@automake@err
\else

```

```

        \renewcommand*{\@gls@doautomake}{}%
    \fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{%
\AtEndDocument{\@gls@doautomake}
\@gls@declareoption{automakegloss}{%
    \glsautomaketrue
    \renewcommand{\gls@automake@option}{automakegloss}%
    \def\gls@automake@val{makegloss}%
    \def\gls@automake@nr{3}%
    \renewcommand{\gls@automake@makegloss}{makeglossaries}%
    \let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%
    \glsautomaketrue
    \renewcommand{\gls@automake@option}{automakeglosslite}%
    \def\gls@automake@val{lite}%
    \def\gls@automake@nr{4}%
    \renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
    \let\@gls@doautomake\@gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
    \ifgls savewrites
        \renewcommand*{\glswritefiles}{\@glswritefiles}%
    \else
        \let\glswritefiles\@empty
    \fi
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
    \ifbool{glscompatible-3.07}{%
        {%
            \@gls@removedoption{compatible-3.07}%
        }%
    }%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
    \ifbool{glscompatible-2.07}{%
        {%
            \@gls@removedoption{compatible-2.07}%
        }%
    }%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%

```

```

\GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@kernelglossredefs@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
\newcommand*{\printsymbols}[1][\@gls@set@symbols@toctitle]{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{}

```

```

\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
  \newcommand*{\newterm}[2][%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
}
\define@choicekey{glossaries.sty}{mfirstuc}{%
  [\gls@mfirstuc@val\gls@mfirstuc@nr]{%
  {expanded,unexpanded}{%
    \ifcase\gls@mfirstuc@nr
      % expanded
      \renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{##1}}%
    \or
      % unexpanded
      \renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{##1}}}%
    \fi
  }
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \@expandtwoargs
    \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@option
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{

```

```

\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
\renewcommand*{\@gls@setacrstyle}{}%
\ifglsacrshortcuts
\def\@gls@setupshortcuts{\glsacrshortcutstrue}%
\else
\def\@gls@setupshortcuts{%
\ifglsacrshortcuts
\DefineAcronymSynonyms
\fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsubentrycounter
\let\@gls@doentrycounterdef\relax
\else
\let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifglsubentrycounter
\let\@gls@dodosubentrycounterdef\relax
\else
\let\@gls@dodosubentrycounterdef\@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@dodosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{
\ifcsundef{chapter}{%
}%
\let\@gls@old@chapter\@chapter
\def\@chapter[#1]#2{\@gls@old@chapter[#1]{#2}%
\ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
}%
}%

```

```

{}
\newcommand*{\@gls@onlypremakeg}{}
\newcommand*{\@onlypremakeg}[1]{%
  \ifx\@gls@onlypremakeg\@empty
    \def\@gls@onlypremakeg{#1}%
  \else
    \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
    \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
  \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
  \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
  \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}
\providecommand*{\glsymbolsgroupname}{Symbols}
\providecommand*{\glsnumbersgroupname}{Numbers}
\newcommand*{\glspluralsuffix}{s}
\newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*{\glsupacrpluralsuffix}{\glsup{t}\glsacrpluralsuffix}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
  \@ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
\newcommand*{\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}[2019/11/30]
  \ifdefvoid\@gls@requested@languages{}%
  {%
    \@for\@gls@thislang:=\@gls@requested@languages\do{%
      \TrackIfKnownLanguage{\@gls@thislang}{}%
      {\GlossariesWarning{unknown language '\@gls@thislang'}}%
    }%
  }%
\fi

```

```

\@gls@usetranslator
\@ifpackageloaded{translator}
{%
  \ifboolexpr
  {
    test {\ifdefstring{\trans@languages}{English}}
    and not
    test {\ifdefstring{bbl@loaded}{english}}
  }
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \usedictionary{glossaries-dictionary}%
    \renewcommand*{\addglossarytocaptions}[1]{%
      \ifcsundef{captions#1}{}%
      {%
        \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
        \expandafter\toks@\expandafter{\@gls@tmp
          \renewcommand*{\glossaryname}{\translate{Glossary}}}%
        }%
        \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
      }%
    }%
  }%
}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
  }%
}

```

```

        \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
}%
\renewcommand*\glossaryname{\translate{Glossary}}%
\renewcommand*\acronymname{\translate{Acronyms}}%
\renewcommand*\entryname{\translate{Notation (glossaries)}}%
\renewcommand*\descriptionname{%
    \translate{Description (glossaries)}}%
\renewcommand*\symbolname{\translate{Symbol (glossaries)}}%
\renewcommand*\pagelistname{%
    \translate{Page List (glossaries)}}%
\renewcommand*\glsymbolsgroupname{%
    \translate{Symbols (glossaries)}}%
\renewcommand*\glsnumbersgroupname{%
    \translate{Numbers (glossaries)}}%
}{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*{@nopostdesc}{%
    \let\org@glspostdescription\glspostdescription
    \def\glspostdescription{%
        \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{@no@post@desc}{\nopostdesc}
\newcommand*\glspar{\par}
\newcommand*\setStyleFile[1]{%
    \renewcommand*\gls@istfilebase{#1}%
    \ifglxindy
        \def\istfilename{\gls@istfilebase.xdy}
    \else
        \def\istfilename{\gls@istfilebase.ist}
    \fi
}
@onlypremakeg\setStyleFile
\ifglxindy
    \def\istfilename{\gls@istfilebase.xdy}
\else
    \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*\gls@istfilebase{\jobname}
\newcommand*\@istfilename[1]{}
\newcommand*\glscompositor{.}
\newcommand*\glsSetCompositor[1]{%
    \renewcommand*\glscompositor{#1}}
@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglxindy
    \newcommand*\glsSetAlphaCompositor[1]{%
        \renewcommand*\@glsAlphacompositor{#1}}
\else

```

```

\newcommand*{\glsSetAlphaCompositor}[1]{%
  \glsnoxindywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{}
\newcommand*{\glsSetSuffixF}[1]{%
  \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{}
\newcommand*{\glsSetSuffixFF}[1]{%
  \renewcommand*{\gls@suffixFF}{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*{\glsnumberformat}[1]{\glsnumber{#1}}%
}
\newcommand{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@currentglossary}}
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \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{\glsglossarymark}[1]{%
    \ifglsucmark
        \mkboth{\glssupercase{#1}}{\glssupercase{#1}}%
    \else
        \mkboth{#1}{#1}%
    \fi
}
}
}
\providecommand{\glossarymark}[1]{%
    \ifglsucmark
        \mkboth{\glssupercase{#1}}{\glssupercase{#1}}%
    \else
        \mkboth{#1}{#1}%
    \fi
}
\newcommand*{\setglossarysection}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossarysection}[2]{%
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname[#1]{#2}%
    }%
    {%
        \csname\@glossarysec\endcsname*{#2}%
        \@gls@toc{#1}{\@glossarysec}%
    }%
    \@glossaryseclabel
}
\newcommand*{\@p@glossarysection}[2]{%
    \glsclearpage
    \phantomsection
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}{\@glossarysec}%
        \csname\@glossarysec\endcsname*{#2}%
    }%
    \@glossaryseclabel
}
}

```

```

\newcommand*\gls@doclearpage}{%
  \ifthenelse{\equal{\@@glossarysec}{chapter}}{%
    {%
      \ifcsundef{cleardoublepage}%
        {%
          \clearpage
        }%
      {%
        \ifcsdef{if@openright}%
          {%
            \if@openright
              \cleardoublepage
            \else
              \clearpage
            \fi
          }%
        {%
          \cleardoublepage
        }%
      }%
    }%
  }%
}
\newcommand*\gls@clearpage}{\gls@doclearpage}
\newcommand*\@gls@toc}[2]{%
  \ifglstoc
    \ifglsnumberline
      \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
      \addcontentsline{toc}{#2}{#1}%
    \fi
  \fi
}
\newcommand*\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@ifinlist
}
\ifglxindy
\newcommand*\@xdycounters{\@glscounter}
\newcommand*\GlsAddXdyCounters[1]{%
\@for\@gls@ctr:=#1\do{%
\edef\@do@addcounter{%
\noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
}%
\noexpand\edef\@do@addcounter{\@xdycounters,%
\noexpand\@gls@ctr}%
}%
\do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnoxindywarning\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"#2#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}

```

```

}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
  \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \edef\@xdyattributes{\@xdyattributes ^~J
    \string"#1\string"}%
  \for\@this@counter:=\@xdycounters\do{%
    \protected@edef\gls@do@addxdyattribute{%
      \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
    }
    \gls@do@addxdyattribute
  }%
  \let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute}
\fi
\ifglsxindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{glsnumber}
  \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]{%
  \glsnoinxindwarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
\newcommand*\@gls@xdy@locationlist}{%
  roman-page-numbers,%
  Roman-page-numbers,%
  arabic-page-numbers,%
  alpha-page-numbers,%
  Alpha-page-numbers,%
  Appendix-page-numbers,%
  arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
  \string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
  :sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
    \string"roman-numbers-lowercase\string"%
  }%
\else
  \expandafter
  \edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
    :sep \string"\@gls@roman\string"%
  }%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
  \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
  \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
  \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
  \string"ALPHA\string"%
}%

```

```

\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
  \string"ALPHA\string"
  :sep \string"@glsAlphacompositor\string"
  \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
  \string"arabic-numbers\string"
  :sep \string"\glscompositor\string"
  \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\GlsAddXdyLocation}[3][]{%
  \def\@gls@tmp{#1}%
  \ifx\@gls@tmp\@empty
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glsopenbrace\string" #3
          :sep \string"\glsclosebrace\string"))
    }%
  \else
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#2\string"^^J\space\space
        \space(:sep "\glsopenbrace"
          #1
          :sep "\glsclosebrace\glsopenbrace" #3
          :sep "\glsclosebrace"))
    }%
  \fi
  \edef\@xdyuserlocationnames{%
    \@xdyuserlocationnames^^J\space\space\space
    \string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
  \glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
  \string"roman-page-numbers\string"^^J\space\space\space
  \string"arabic-page-numbers\string"^^J\space\space\space
  \string"arabic-section-numbers\string"^^J\space\space\space

```

```

\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoxywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
\expandafter\toks@\expandafter{\@xdysortrules}%
\protected@edef\@xdysortrules{\the\toks@ ^^J
(sort-rule \string"#1\string" \string"#2\string")}%
}
\else
\newcommand*\GlsAddSortRule[2]{%
\glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
\def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
\newcommand*\GlsAddXdyStyle[1]{%
\edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
\newcommand*\GlsAddXdyStyle[1]{%
\glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
\newcommand*\GlsSetXdyStyles[1]{%
\edef\@xdyrequiredstyles{#1}}
\else
\newcommand*\GlsSetXdyStyles[1]{%
\glsnoxywarning\GlsSetXdyStyles}
\fi
\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][1]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\@GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\gls@codepage\@empty
      \@ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\@GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifgls@xindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glsnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space#2)}%
  }%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@gls@acronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forglsentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
  \@for#2:=\@glo@list\do

```

```

    {%
      \ifdefempty{#2}{-}{#3}%
    }%
  }
\newcommand*{\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
  {%
    \forglsentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glo@list@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
  }
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{-}{#2}%
  }
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}%
    #3%
  }%
}

```

```

}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glstoifexists{#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
  }%
}
\newcommand{\ifglshasparent}[3]{%
  \glstoifexists{#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}%
    }%
}
}
\newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo\@glsdetoklabel{#2}@#1}{#3}{#4}}
\newrobustcmd*{\ifglshasfield}[4]{%
    \glsdoifexists{#2}%
    {%
        \letcs{\@glo@thisvalue}{glo\@glsdetoklabel{#2}@#1}%
        \ifdef\@glo@thisvalue
        {%
            \ifdefempty\@glo@thisvalue
            {%
                #4%
            }%
            {%
                \ifdefequal\@glo@thisvalue\@gls@default@value
                {%
                    #4%
                }%
                {%
                    \let\glscurrentfieldvalue\@glo@thisvalue
                    #3%
                }%
            }%
        }%
        {%
            \@gls@fetchfield{\@gls@thisfield}{#1}%
            \ifdef\@gls@thisfield

```



```

    {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 @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}
\@onlypremakegnewglossary
\newcommand*\@newglossary[4]{
\def\@gls@setcounter#1[#2]{%

```

```

\expandafter\def\csname @glo@#1@counter\endcsname{#2}%
\ifglxindy
  \GlsAddXdyCounters{#2}%
\fi
}
\newcommand*{\@gls@getcounter}[1]{%
\csname @glo@#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{glo@list@#1}{,}%
\ifcsundef{gls@#1@entryfmt}%
{%
\defglsentryfmt[#1]{\glsentryfmt}%
}%
{}}%
\ifdefempty\@gls@nohyperlist
{%
\renewcommand*{\@gls@nohyperlist}{#1}%
}%
{%
\protected@eappto\@gls@nohyperlist{,#1}%
}%
}
\newcommand*{\@ignored@glossaries}{}
\newcommand*{\ifignoredglossary}[3]{%
\protected@edef\@gls@igtype{#1}%
\expandafter\DTLifinlist\expandafter
{\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
}

```

```

\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*\@gls@set@xr@key}[3]{%
\renewcommand*\@gls@xr@key}{#1}%
\gls@checkseeallowed
\def#2{#3}%
\@glo@seeautonumberlist
}
\newcommand*\@gls@xr@key}{see}
\newcommand*\@gls@checkseeallowed}{%
\@gls@see@noindex
}

```

```

\newcommand*\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {'\gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
[\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \gls@savenonumberlist{false}%
  \fi
}
\newcommand*\gls@noidx@nonumberlist}[1]{%
  \cslet\glo@\@glo@label @prenumberlist\@glo@prefix
}%
\newcommand{\glsnoidxprenumberlist}[1]{%
  \csuse\glo@\glsdetoklabel{#1}@prenumberlist}
\newcommand*{\@gls@savenonumberlist}[1]{%
\newcommand*{\@gls@initnonumberlist}{%
\newcommand*{\@gls@storenonumberlist}[1]{%
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet\glo@\glsdetoklabel{##1}@nonumberlist}\@glo@nonumberlist}%
    }%
    }%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
}

```

```

\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
  \GlossariesWarning{Missing number list for entry '#1'.
  Maybe makeglossaries + rerun required}%
\else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
}

```

```

\fi
}
\newcommand*\@glsdefaultsort{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}
}
}

```

```

    }%
  }%
}%
{
  \ifdefequal{#4}{\@gls@default@value}%
  {
    \@gls@expand@field{#2}{#3}{#1}%
  }%
  {
    \@gls@expand@field{#2}{#3}{#4}%
  }%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expandfields{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpandfields{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*\gls@defdocnewglossaryentry{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile

```

```

\immediate\openout\@gls@deffile=\jobname.glsdefs
}%
{}%
\ifglsentryexists{#1}%
{%
\xifinlist{#1}{\gls@docdef@labellist}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in the
document environment (ignoring new definition)}%
}%
{%
\xifinlist{#1}{\gls@glsdefs@labellist}%
{%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in
the preamble (in which case remove the duplicate definition)
or you have just updated to glossaries.sty
v4.47 (in which case delete ‘\jobname.glsdefs’)}%
}%
}%
}%
{%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
\@gls@enablesavenonumberlist
\edef\@gls@restreat{\noexpand\catcode‘\noexpand\@=\number\catcode‘\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{-}{-}%
\@gls@restreat
\undef\@gls@restreat
\gls@defdocnewglossaryentry
}
\newcommand*\glsdefs@newdocentry}[2]{%
\ifglsentryexists{#1}%
{\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
(if you have moved the definition there from the document
environment, delete \jobname.glsdefs)}}%
{%
\gls@glsdefs@addlabel{#1}%
\gls@defglossaryentry{#1}{#2}%
}
}

```

```

}%
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
        \glspercentchar
      }%
    }%
  }%
  }%
  \gls@writedefhook
  \immediate\write\@gls@deffile
  {%
    \glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar^^J%
  }%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%

```

```

{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
\edef\@gls@thisval{#2}%
\@for\@gls@map:=\@gls@keymap\do{%
\edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
\ifdefequal{\@this@key}{\@gls@thisval}%
{%
\edef#1{\expandafter\@secondoftwo\@gls@map}%
\@endfortrue
}%
}%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
}%
\PackageError{glossaries}{Key ‘#1’ already exists}{%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter

```

```

        {\csname gls@assign@#1@field\endcsname}[2]{%
          \@gls@expand@field{##1}{#1}{##2}%
        }%
      }%
    {}%
  \@gls@saddkey{#1}%
}
\newcommand*{\@gls@saddkey}[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}{\glsuppercase{#3{##2}##3}}%
    }%
    \newrobustcmd*{#7}{%
      \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
    }%
    \glsmfuaddmap{#3}{#4}%
    \glsmfuaddmap{#5}{#6}%
    \glsmfublocker{#7}%
  }%
  {%
    \PackageError{glossaries}{Key '#1' already exists}{}%
  }%
}
\newrobustcmd{\glsfieldxdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Field '#2' doesn't exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldedef}[3]{%
  \glsdoifexists{#1}%
  {%

```

```

\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{
  \protected@csedef{glo@\@glo@label @#2}{#3}%
}%
{
  \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}
}
\newrobustcmd{\glsfieldgdef}[3]{%
\glsdoifexists{#1}%
{
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {
    \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
}
\newrobustcmd{\glsfieldddef}[3]{%
\glsdoifexists{#1}%
{
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {
    \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
  }%
  {
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
}
\newrobustcmd{\glsfieldfdef}[3]{%
\glsdoifexists{#1}%
{
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
}
\newcommand*\glsunexpandedfieldvalue}[2]{%

```

```

\ifcsname glo@glstetoklabel{#1}@#2\endcsname
\expandafter\expandonce\csname glo@glstetoklabel{#1}@#2\expandafter\endcsname
\fi
}
\newrobustcmd{\ifglstetok}[5]{%
\glstetokexists{#1}%
{%
\protected@edef\glo@label{\glstetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstring{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Field '#2' doesn't exist}{}%
}%
}%
}
\newrobustcmd{\ifglstetokdef}[5]{%
\glstetokexists{#1}%
{%
\protected@edef\glo@label{\glstetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\expandafter\ifdefstrequal\csname glo@\glo@label @#2\endcsname{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Field '#2' doesn't exist}{}%
}%
}%
}
\newcommand{\ifglstetokcseq}[5]{%
\glstetokexists{#1}%
{%
\protected@edef\glo@label{\glstetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key '#2' doesn't exist}{}%
}%
}%
}
\newcommand*\glswritedefhook{}
\newcommand*\glsassign@desc[1]{%
\glsassign@field{#1}{desc}{\glo@desc}%
\glsassign@field{\glo@desc}{#1}{descplural}{\glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glstetoknoexists{#1}%

```

```

{%
  \bgroup
  \let\@org@newglossaryentryprehook\@newglossaryentryprehook
  \long\def\@newglossaryentryprehook{%
    \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
    \@org@newglossaryentryprehook
  }%
  \renewcommand*\@gls@assign@desc}[1]{%
    \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
    \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
  }
  \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@gls@name
  \let\@glo@desc\@gls@desc
  \let\@glo@descplural\@gls@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
  \def\@glo@parent{}%
  \def\@glo@prefix{}%
  \@gls@initnonumberlist
  \def\@glo@useri{}%
  \def\@glo@userii{}%
  \def\@glo@useriii{}%
  \def\@glo@useriv{}%
  \def\@glo@userv{}%
  \def\@glo@uservi{}%
  \def\@glo@short{}%
  \def\@glo@shortpl{}%
  \def\@glo@long{}%
}

```

```

\def\@glo@longpl{}%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glolist@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
{}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\empty
  \fi
}%
{}%
{}%
{}%
\protected@edef\@glolist@{\csname glolist@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glolist@\@glo@type\endcsname{%
  \@glolist@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
    }%
  }%
\fi

```

```

\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\ifftrue
}

```

```

}%
\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@glsee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
      \do@glsee
    }%
    \@glo@autoseehook
  }%
}
\newcommand*{\@glo@autoseehook}{%
}
\newcommand*{\@newglossaryentryprehook}{%
}
\newcommand*{\@newglossaryentryposthook}{%
}
\newcommand*{\@newglossaryentry@defcounters}{%
}
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\glo@list{,%
}
\for@glseentries[\glo@type]{\glo@label}%
  {%
    \ifdequal\@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}%
  \ifglsxindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\@glo@parent\@empty
      \@onelevel@sanitize\@glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossaryentryfield{\@glo@esclabel}%
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
        \@glo@sort\@gls@actualchar\@glo@prefix
        \@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
      }%
    \fi
  \fi
}
\newcommand*{\@gls@if@measuring}[2]{%
  \ifmeasuring@
    #1%
  \else
    #2%
  \fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand{\@gls@if@measuring}[2]{#2}
\AtBeginDocument{%

```

```

\@ifpackageloaded{amsmath}%
{\let\gls@if@measuring\@gls@if@measuring}%
}%
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter}[1]{%
\ifcsdef{c#1}%
{%
\epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
\expandafter\advance\csname c@#1\endcsname by \@ne
}%
}%
}
\newcommand{\gls@start@measuring}{%
\ifx\glsifmeasuring\@firstoftwo
\else
\let\gls@org@target\glstarget
\let\gls@org@link\glslink
\def\gls@measuring@reset{}%
\fi
\let\glsifmeasuring\@firstoftwo
\let\refstepcounter\gls@local@stepcounter
\let\label\@gobble
\let\glstarget\@secondoftwo
\let\@glslink\@secondoftwo
}
\newcommand{\gls@stop@measuring}{%
\renewcommand*\glsifmeasuring{\gls@if@measuring}%
\let\refstepcounter\gls@org@refstepcounter
\let\label\gls@org@label
\let\glstarget\gls@org@target
\let\@glslink\gls@org@link
\gls@measuring@reset
\def\gls@measuring@reset{}%
}
\newcommand{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}

```

```

\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\glsunset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsunset{#1}%
}%
}%
}
\newcommand*\glslocalunset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalunset{#1}%
}%
}%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
\expandafter\let\csname ifglo@\glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%

```

```

\expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*{\glsenableentrycount}{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*\newglossaryentry[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}

```

```

}%
\newcommand*\glentrycurrcount}[1]{%
  \ifcsundef{glo\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glentryprevcount}[1]{%
  \ifcsundef{glo\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \ifglsresetcurrcount\csgdef{glo\glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo\glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgls@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cglspl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglsplformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@glspl@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@cGls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGlspl@##1##2[##3]{%

```

```

\ifnum\glsentryprevcount{##2}=1\relax
  \cGlsplformat{##2}{##3}%
  \glsunset{##2}%
\else
  \@Glspl@{##1}{##2}[##3]%
\fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
    {}%
  }%
}
\newcommand*{\@gls@entry@count}[2]{}
\newrobustcmd*{\cgl}{\@gls@hyp@opt@cgl}
\newcommand*{\@cgl}[2][{}]{%
  \new@ifnextchar[{\@cgl@{##1}{##2}}{\@cgl@{##1}{##2}[]}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl}\space is defaulting to
  \string\gls\space since you haven't enabled entry counting}%
  \@gls@{##1}{##2}[##3]%
}
\newcommand*{\cglformat}[2]{%
  \ifglshaslong{##1}{\glsentrylong{##1}}{\glsentryfirst{##1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt@cGls}
\glsmfuaddmap{\cgl}{\cGls}
\newcommand*{\@cGls}[2][{}]{%
  \new@ifnextchar[{\@cGls@{##1}{##2}}{\@cGls@{##1}{##2}[]}%
}

```

```

\def\cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]%
}
\newcommand*\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*\cGlspl{\@gls@hyp@opt\@cGlspl}
\newcommand*\cGlspl}[2][{}]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}}[{}]}%
}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newrobustcmd*\cGlspl{\@gls@hyp@opt\@cGlspl}
\glsmfuaddmap{\cGlspl}{\cGlspl}
\newcommand*\cGlspl}[2][{}]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}}[{}]}%
}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \@Glspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glstextformat}[1]{#1}
\newcommand*\glsentryfmt{\glsgenentryfmt}
\newcommand*\glsgenentryfmt{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%

```

```

    \glsentryplural{\glslabel}\glsinsert
  }%
  {%
    \glsentryfirstplural{\glslabel}\glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \Glsentryplural{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirstplural{\glslabel}\glsinsert
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \glsuppercase{\glsentryplural{\glslabel}\glsinsert}%
  }%
  {%
    \glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glsentrytext{\glslabel}\glsinsert
    }%
    {%
      \glsentryfirst{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirst{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
  }%
}

```



```

        as its value%
    }%
}%
{%
    \def\@gls@counter{#1}%
}%
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef\hyperlink{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
    \let@if@org@KV@glslink@local@ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
    \ifKV@glslink@local
        \let\@gls@do@glsunset@glslocalunset
    \else
        \let\@gls@do@glsunset@glsunset
    \fi
    \let@ifKV@glslink@local@if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\@glslinkvar}[3]{#1}
\newcommand*\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*\s@gls@hyp@opt}[1][ ]{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*\p@gls@hyp@opt}[1][ ]{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*\glslink{%
    \@gls@hyp@opt\@gls@@link
}
\newcommand*\@gls@@link}[3][ ]{%
    \glsdoifexistsordo{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\relax
        \@gls@link[#1]{#2}{#3}%
    }-%
    \glstextformat{#3}%
}%

```

```

\glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@link
}
\newcommand*{\@Gls@link}[3][{}]{%
  \@gls@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glsstype
    \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{\glsstype}{\@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\glsstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
}

```

```

\glslinkpostsetkeys
\@gls@saveentrycounter
\@gls@setsort{\glslabel}%
\glsencapwrcontent{\do@wrglossary{#2}}%
\ifKV@glslink@hyper
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
\@gls@restore@glslocal
}
\newcommand*{\glsencapwrcontent}[1]{#1}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}%
  {
    \ifcsundef{xatlevel@}%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    {%
      \ifx\xatlevel@\@empty
        \edef\theglsentrycounter{\expandafter\noexpand
          \csname the\@gls@counter\endcsname}%
      \else
        \savecounters@
        \advance\c@equation by 1\relax
        \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
        \ifcsundef{theH\@gls@counter}%
        {%
          \def\@gls@Hcounter{\theglsentrycounter}%
        }%
        {%
          \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
        }%
        \protected@edef\theHglentrycounter{\@gls@Hcounter}%
        \restorecounters@
      \fi
    }%
  }%
  \edef\theglsentrycounter{\expandafter\noexpand
    \csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
  \ifcsundef{theH\@gls@counter}%

```

```

    {%
      \def\theHglentrycounter{\theHglentrycounter}%
    }%
    {%
      \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}%
  \ifglsesclocations

```

```

\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
  \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
  \@onelevel@sanitize\@gls@sanitized@tmp
  \edef\gls@dostsubst{%
    \noexpand\DTLsubstituteall\noexpand\gls@xdystring
    {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
  }%
  \gls@dostsubst
}%
\fi
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checkescquote#1\@nil\""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checkescactual#1\@nil\?\?\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checkactual#1\@nil??\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checkbar#1\@nil||\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checkeschar#1\@nil\|\|\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checklevel#1\@nil!!\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@gls@checkedmkidx{%
      \expandafter\@gls@checkescllevel#1\@nil\!\!\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    }\@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
\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
\fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%

```

```

\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesclevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%

```

```

\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\bgroup
\catcode'\|0\relax
\catcode'\|12\relax
\gdef|\@gls@checkesc#1\#2\#3\null{%
|\@gls@tmpb=\expandafter{|\@gls@checkedmkidx}%
|toks@={#1}%
|ifx|null#2|null
|ifx|null#3|null
|edef|\@gls@checkedmkidx{|the|\@gls@tmpb|the|toks@}%
|def|\@gls@checkesc{|relax}%

```

```

|else
|edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
|@gls@quotechar|@backslashchar|@gls@quotechar|@backslashchar}%
|def|@@gls@checkesc{|@gls@checkesc#3|null}%
|fi
|else
|edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
|@gls@quotechar|@backslashchar}%
|ifx|null#3|null
|def|@@gls@checkesc{|@gls@checkesc#2\\|null}%
|else
|def|@@gls@checkesc{|@gls@checkesc#2\\#3|null}%
|fi
|fi
|@@gls@checkesc
}
|egroup
\def|@gls@xdycheckquote#1"#2"#3\null{%
|@gls@tmpb=\expandafter{|@gls@checkedmkidx}%
|toks@=#1}%
|ifx\null#2\null
|ifx\null#3\null
|edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@}%
|def|@@gls@xdycheckquote{|relax}%
|else
|edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
|string\"string\"}%
|def|@@gls@xdycheckquote{|@gls@xdycheckquote#3\null}%
|fi
|else
|edef|@gls@checkedmkidx{|the|@gls@tmpb|the|toks@
|string\"}%
|ifx\null#3\null
|def|@@gls@xdycheckquote{|@gls@xdycheckquote#2\"}\null}%
|else
|def|@@gls@xdycheckquote{|@gls@xdycheckquote#2"#3\null}%
|fi
|fi
|@@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand|@gls@xdycheckbackslash##1|@backslashchar
##2|@backslashchar##3\noexpand\null{%
\noexpand|@gls@tmpb=\noexpand\expandafter
{|noexpand|@gls@checkedmkidx}%
\noexpand\toks@=#1}%
\noexpand|ifx\noexpand\null##2\noexpand\null
\noexpand|ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand|@gls@checkedmkidx{%
\noexpand|the\noexpand|@gls@tmpb\noexpand|the\noexpand\toks@}%

```

```

\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength@gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
\glsdoshowtarget{#1}%
{%
\glsmeasureheight{gls@tmplen}{#2}%
\raisebox{gls@tmplen}{\hypertarget{#1}{}}%
#2%
}%
}
\newcommand*\glsdohyperlink}[2]{%
\glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
\let@glslink@glsdonohyperlink
}%
{%
\let@glslink@glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
\let@glstarget\@secondoftwo

```

```

}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2] [] {%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} []}]%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{gls}{\Gls}
\newcommand*{\@Gls}[2] [] {%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}]%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
}

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2] [] {%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2] [] {%
  \new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2} []}%
}
\def\@glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glspl}{\Glspl}
\newcommand*{\@Glspl}[2] [] {%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2} []}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo

```

```

\let\glscapscase\@secondofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
\new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][{}]{%
\glsdoifexists{#2}{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\def\glscustomtext{#3}%
\def\glsinsert{}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
\@glsdisp[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glsdisp}{\Glsdisp}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
\glsdoifexists{#2}%

```

```

{%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \@gl@link[#1]{#2}{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glstext}{\@gl@hyp@opt\@glstext}
\newcommand*{\@glstext}[2] []{%
  \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2} []}}
\def\@glstext@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\glstentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gl@hyp@opt\@GLStext}
\glsmfublocker{\GLStext}
\newcommand*{\@GLStext}[2] []{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2} []}}
\def\@GLStext@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\glssuppercase{\glstentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gl@hyp@opt\@Glstext}
\glsmfuaddmap{\glstext}{\Glstext}
\newcommand*{\@Glstext}[2] []{%
  \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2} []}}
\def\@Glstext@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
}
\newrobustcmd*{\glfirst}{\@gl@hyp@opt\@glfirst}
\newcommand*{\@glfirst}[2] []{%
  \new@ifnextchar[{\@glfirst@{#1}{#2}}{\@glfirst@{#1}{#2} []}}
\def\@glfirst@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\glstentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gl@hyp@opt\@Glsfirst}
\glsmfuaddmap{\glfirst}{\Glsfirst}
\newcommand*{\@Glsfirst}[2] []{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}}
\def\@Glsfirst@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gl@hyp@opt\@GLSfirst}
\glsmfublocker{\GLSfirst}
\newcommand*{\@GLSfirst}[2] []{%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2} []}}
\def\@GLSfirst@#1#2[#3]{%
  \@gl@field@link{#1}{#2}{\glssuppercase{\glstentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gl@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] []{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}}
\def\@glsplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2][\%
    \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}]
\def\@Glsplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2][\%
    \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2}[]}]
\def\@GLSplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][\%
    \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}]
\def\@glsfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\glsmfublocker{\GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2][\%
    \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2}[]}]
\def\@GLSfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsupercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2][\%
    \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2}[]}]
\def\@glsname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\glsmfuaddmap{\glsname}{\Glsname}
\newcommand*{\@Glsname}[2][\%
    \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2}[]}]
\def\@Glsname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}

```

```

\glsmfublocker{\GLSname}
\newcommand*{\@GLSname}[2] [] {%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}]
\def\@GLSname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] [] {%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}]
\def\@glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\glsmfuaddmap{\glsdesc}{\Glsdesc}
\newcommand*{\@Glsdesc}[2] [] {%
  \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2} []}]
\def\@Glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\glsmfublocker{\GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}]
\def\@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}]
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural}{\@gls@hyp@opt\@Glsdescplural}
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar[{\@Glsdescplural@{#1}{#2}}{\@Glsdescplural@{#1}{#2} []}]
\def\@Glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\glsmfublocker{\GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}]
\def\@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\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}
\glsmfuaddmap{\glsymbol}{\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}
\glsmfublocker{\GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
    \new@ifnextchar [{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}]
\def\@GLSsymbol@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glsymbolplural}{\@gls@hyp@opt\@glsymbolplural}
\newcommand*{\@glsymbolplural}[2] [] {%
    \new@ifnextchar [{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}]
\def\@glsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfuaddmap{\glsymbolplural}{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
    \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
    \new@ifnextchar [{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
\def\@GLSsymbolplural@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
    \new@ifnextchar [{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
\def\@glsuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfuaddmap{\glsuseri}{\GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
    \new@ifnextchar [{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3] {%
    \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}

```

```

\glsmfublocker{\GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2] [] {%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2} []}]
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\glsmfuaddmap{\glsuserii}{\Glsuserii}
\newcommand*{\@Glsuserii}[2] [] {%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2} []}]
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar[{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2][\%
    \new@ifnextchar[{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2}[]}]
\def\@Glsuseriv@#1#2[#3]{\%
    \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2][\%
    \new@ifnextchar[{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2}[]}]
\def\@GLSuseriv@#1#2[#3]{\%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2][\%
    \new@ifnextchar[{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2}[]}]
\def\@glsuserv@#1#2[#3]{\%
    \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2][\%
    \new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}[]}]
\def\@GLSuserv@#1#2[#3]{\%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2][\%
    \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}[]}]
\def\@glsuservi@#1#2[#3]{\%
    \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfuaddmap{\glsuservi}{\Glsuservi}
\newcommand*{\@Glsuservi}[2][\%
    \new@ifnextchar[{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2}[]}]
\def\@Glsuservi@#1#2[#3]{\%
    \@gls@field@link{#1}{#2}{\Glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}

```

```

\glsmfublocker{\GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}]
\def\@GLSuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}]%
}
\def\@acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\gls label{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
\def\@ACRshort#1#2[#3] {%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
  \def\glslabel{#2}%
  \let\glsifplural\@secondoftwo
  \let\gls caps case\@thirdofthree
  \let\glsinsert\@empty
  \def\gls custom text{%
    \gls upper case{\acronymfont{\glsentryshort{#2}}#3}%
  }%
  \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
}%
\gls post link hook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2][{}]{%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2}[]]}%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
  \gls post link hook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\glsmfuadddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2][{}]{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]]}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls type @entryfmt\endcsname}%
  }%
}

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2] []{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}]%
}
\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
\newcommand*{\ns@acrlong}[2] []{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}]%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\@ns@Acrlong}
\glsmfuaddmap{\Acrlong}{\Acrlong}
\newcommand*{\ns@Acrlong}[2] []{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2} []}]%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\gl@ifplural\@secondoftwo
\let\gl@scapscase\@secondofthree
\let\gl@insert\@empty
\def\glscustomtext{%
  \Gl@entrylong{#2}#3%
}%
\@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\gl@postlinkhook
}
\newrobustcmd*{\ACRlong}{\@gl@hyp@opt\@ns@ACRlong}
\gl@mfublocker{\ACRlong}
\newcommand*{\ns@ACRlong}[2][{}]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}}[{}]}%
}
\def\@ACRlong#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\gl@ifplural\@secondoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@insert\@empty
    \def\glscustomtext{%
      \gl@supercase{\gl@entrylong{#2}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gl@hyp@opt\@ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}}[{}]}%
}
\def\@acrlongpl#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\gl@ifplural\@firstoftwo
    \let\gl@scapscase\@firstofthree
    \let\gl@insert\@empty
    \def\glscustomtext{%
      \gl@entrylongpl{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \gl@postlinkhook
}

```

```

}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}{%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}{%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%

```

```

\letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
\ifdef\@glo@text
{%
  \expandafter\glsentencecase\expandafter{\@glo@text}%
}%
{%
  ??\PackageError{glossaries}{The field '#2' doesn't exist for glossary
  entry '\glsdetoklabel{#1}'}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%
}%
??%
}%
}
\newcommand{\@Glsentryfield}[2]{%
  \glsTeXorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
\ifdef\TeXorpdfstring
{\newcommand{\glsTeXorpdfstring}{\TeXorpdfstring}}
{\newcommand{\glsTeXorpdfstring}[2]{#1}}
\newcommand*{\glsentryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \glsTeXorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{name}}}%
}
\glsmfuaddmap{\glsentryname}{\Glsentryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
    \expandafter\glsentencecase\expandafter{\@glo@text}%
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@Gls@getbody\@glo@text{\@nil
    \expandafter\ifx\@Gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@Gls@rest
    \else
    \expandafter\ifx\@Gls@body\glsentryshort\relax

```

```

\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\glsentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Glsentryfield{#1}{desc}}
\glsmfuaddmap{\glsentrydesc}{\Glsentrydesc}
\newcommand*{\glsentrydescplural}[1]{%
\@gls@entry@field{#1}{descplural}%
}
\newcommand*{\Glsentrydescplural}[1]{%
\@Glsentryfield{#1}{descplural}%
}
\glsmfuaddmap{\glsentrydescplural}{\Glsentrydescplural}
\newcommand*{\glsentrytext}[1]{\@gls@entry@field{#1}{text}}
\newcommand*{\Glsentrytext}[1]{%
\@Glsentryfield{#1}{text}%
}
\glsmfuaddmap{\glsentrytext}{\Glsentrytext}
\newcommand*{\glsentryplural}[1]{%
\@gls@entry@field{#1}{plural}%
}
\newcommand*{\Glsentryplural}[1]{%
\@Glsentryfield{#1}{plural}%
}
\glsmfuaddmap{\glsentryplural}{\Glsentryplural}
\newcommand*{\glsentrysymbol}[1]{%
\@gls@entry@field{#1}{symbol}%
}
\newcommand*{\Glsentrysymbol}[1]{%

```

```

    \@Glsentryfield{#1}{symbol}%
}
\glsmfuaddmap{\glsentrysymbol}{\Glsentrysymbol}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\Glsentrysymbolplural}[1]{%
  \@Glsentryfield{#1}{symbolplural}%
}
\glsmfuaddmap{\glsentrysymbolplural}{\Glsentrysymbolplural}
\newcommand*{\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newcommand*{\Glsentryfirst}[1]{%
  \@Glsentryfield{#1}{first}%
}
\glsmfuaddmap{\glsentryfirst}{\Glsentryfirst}
\newcommand*{\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newcommand*{\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
\glsmfuaddmap{\glsentryfirstplural}{\Glsentryfirstplural}
\newcommand*{\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
\newcommand*{\glsentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*{\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuseri}[1]{%

```

```

    \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glentryuseri}{\Glsentryuseri}
\newcommand*{\glentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
}
\newcommand*{\Glsentryuserii}[1]{%
    \@Glsentryfield{#1}{userii}%
}
\glsmfuaddmap{\glentryuserii}{\Glsentryuserii}
\newcommand*{\glentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
}
\newcommand*{\Glsentryuseriii}[1]{%
    \@Glsentryfield{#1}{useriii}%
}
\glsmfuaddmap{\glentryuseriii}{\Glsentryuseriii}
\newcommand*{\glentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
}
\newcommand*{\Glsentryuseriv}[1]{%
    \@Glsentryfield{#1}{useriv}%
}
\glsmfuaddmap{\glentryuseriv}{\Glsentryuseriv}
\newcommand*{\glentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
}
\newcommand*{\Glsentryuserv}[1]{%
    \@Glsentryfield{#1}{userv}%
}
\glsmfuaddmap{\glentryuserv}{\Glsentryuserv}
\newcommand*{\glentryuservi}[1]{%
    \@gls@entry@field{#1}{uservi}%
}
\newcommand*{\Glsentryuservi}[1]{%
    \@Glsentryfield{#1}{uservi}%
}
\glsmfuaddmap{\glentryuservi}{\Glsentryuservi}
\newcommand*{\glentryshort}[1]{\@gls@entry@field{#1}{short}}
\newcommand*{\Glsentryshort}[1]{%
    \@Glsentryfield{#1}{short}%
}
\glsmfuaddmap{\glentryshort}{\Glsentryshort}
\newcommand*{\glentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newcommand*{\Glsentryshortpl}[1]{%
    \@Glsentryfield{#1}{shortpl}%
}
\glsmfuaddmap{\glentryshortpl}{\Glsentryshortpl}
\newcommand*{\glentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%

```

```

    \@Glsentryfield{#1}{long}%
}
\glsmfuaddmap{\glsentrylong}{\Glsentrylong}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
    \@Glsentryfield{#1}{longpl}%
}
\glsmfuaddmap{\glsentrylongpl}{\Glsentrylongpl}
\newcommand*{\glsentryfull}[1]{%
    \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{#1} (\glsentryshort{#1})%
}
\glsmfuaddmap{\glsentryfull}{\Glsentryfull}
\newcommand*{\glsentryfullpl}[1]{%
    \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}
\glsmfuaddmap{\glsentryfullpl}{\Glsentryfullpl}
\newcommand*{\glsentrynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
        \@gls@entry@field{#1}{numberlist}%
    }%
}
\@ifpackageloaded{hyperref} {%
    \newcommand*{\glsdisplaynumberlist}[1]{%
        \GlossariesWarning
        {%
            \string\glsdisplaynumberlist\space
            doesn't work with hyperref.^^JUsing
            \string\glsentrynumberlist\space instead%
        }%
        \glsentrynumberlist{#1}%
    }%
}%
{%
    \newcommand*{\glsdisplaynumberlist}[1]{%
        \glsdoifexists{#1}%
        {%
            \bgroup
            \protected@edef\@glo@label{\glsdetoklabel{#1}}%
            \let\@org@glsnumberformat\glsnumberformat
            \def\glsnumberformat##1{##1}%
            \protected@edef\the@numberlist{%
                \csname glo@\@glo@label @numberlist\endcsname}%
            \def\@gls@numlist@sep{}%

```

```

\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*\@glsnumlistsep{, }
\newcommand*\@glsnumlistlastsep{ \& }
\newcommand*\@gls@hyperlink}[2][\@gls@entrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\@glo@linkprefix\@gls@detoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@gls@numberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\@glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@gls@numberformat{\@gls@numberformat}%
    \protected@edef\@gls@counter{\csname glo@\@gls@detoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
}
\glsmfuexcl{\@glsadd}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}
\newrobustcmd*\@glsaddall}[1][{}]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
}

```

```

\forallglsentries[\@glo@type]{\@glo@entry}{%
  \glsadd[#1]{\@glo@entry}%
}%
}
\newrobustcmd*\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore}[1]{
\edef\glsopenbrace{\expandafter@gobble\string\{ }
\edef\glsclosebrace{\expandafter@gobble\string\} }
\edef\glsbackslash{\expandafter@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter}{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits){%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \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*{\@glsminrange}{2}
\ifglsxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
  \def\writeist{%
    \ifundef{\glswrite}{\newwrite\glswrite}{}%
    \@gls@addpredefinedattributes
    \openout\glswrite=\istfilename
    \write\glswrite{;; xindy style file created by the glossaries
      package}%
    \write\glswrite{;; for document '\jobname' on
      \the\year-\the\month-\the\day}%
    \write\glswrite{^^J; required styles^^J}
    \@for\@xdystyle:=\@xdyrequiredstyles\do{%
      \ifx\@xdystyle\@empty
      \else
        \protected@write\glswrite{{(require
          \string"\@xdystyle.xdy\string")}}%
      \fi
    }%
    \write\glswrite{^^J%
      ; list of allowed attributes (number formats)^^J}%
    \write\glswrite{(define-attributes ((\@xdyattributes)))}%
    \write\glswrite{^^J; user defined alphabets^^J}%
    \write\glswrite{\@xdyuseralphabets}%
    \write\glswrite{^^J; location class definitions^^J}%
    \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
      \protected@write\glswrite{{(define-location-class
        \string"\@gls@classI\string"^^J\space\space\space
        (
          :sep "{ }{"
          \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
          :sep "}"
        )
        ^^J\space\space\space
        :min-range-length \@glsminrange^^J%
      )
    }%
  }%

```

```

{%
  \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{}\{(define-location-class
      \string"\@gls@classII-\@gls@classI\string"
      ^^J\space\space\space
      (
        :sep "{"
        \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
        :sep "}{"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
    )
  }%
}%
}
}
}
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glsseeformat\string"
  :close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
  {%
    \@for\@this@attr:=\@xdyattributelist\do{%
      \protected@write\glswrite{}\{\string\providecommand*%
        \expandafter\string
        \csname glsX\@this@ctr X\@this@attr\endcsname[2]%
        {%
          \string\setentrycounter
            [\expandafter\@gobble\string\#1]{\@this@ctr}%
          \expandafter\string
          \csname\@this@attr\endcsname
            {\expandafter\@gobble\string\#2}%
        }%
      }%
    }%
  }%
}

```

```

    }%
  }%
}%
\write\glswrite{%
  \string\begin
  {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
  \space\space:close \string"\glpercentchar\glstildechar n\string
  \end{theglossary}\string\glossarypostamble
  \glstildechar n\string" ^^J\space\space\space
  :tree)}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}

```

```

\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\glspercentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspercentchar\space for document
'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@gls@actualchar'}
\write\glswrite{encap '@gls@encapchar'}
\write\glswrite{level '@gls@levelchar'}
\write\glswrite{quote '@gls@quotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string"}
\write\glswrite{preamble \string\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}
\write\glswrite{postamble \string\string%\string\n\string
\end{theglossary}\string\glossarypostamble\string\n
\string"}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
\string"}
\write\glswrite{item_0 \string\string%\string\n\string"}
\write\glswrite{item_1 \string\string%\string\n\string"}
\write\glswrite{item_2 \string\string%\string\n\string"}
\write\glswrite{item_01 \string\string%\string\n\string"}
\write\glswrite{item_x1
\string\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string\string%\string\n\string"}
\write\glswrite{item_x2
\string\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string\string\{\string
\glossaryentrynumbers\string\}\string\relax \string"}
\write\glswrite{delim_1 \string\string\{\string
\glossaryentrynumbers\string\}\string\relax \string"}
\write\glswrite{delim_2 \string\string\{\string
\glossaryentrynumbers\string\}\string\relax \string"}

```

```

\write\glswrite{delim_t \string\string\}\string\}\string"}
\write\glswrite{delim_n \string\string\delimN \string"}
\write\glswrite{delim_r \string\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\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@checkesquote####1\noexpand\@nil
  \expandonce{\curname#1\endcurname}\expandonce{\curname#1\endcurname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkesactual####1\noexpand\@nil
  \noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
  \noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked

```

```

\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@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}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkescllevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi

```

```

\expand\else
\expand\edef\expand\@gls@checkedmkidx{%
\expand\the\expand\@gls@tmpb\expand\the\expand\toks@
\expand\@gls@quotechar\expand\string
\expandonce{\csname#1\endcsname}\expand\@gls@quotechar}%
\expand\ifx\expand\null####3\expand\null
\expand\def\expand\@gls@checkescquote{%
\expand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\expand\null}%
\expand\else
\expand\def\expand\@gls@checkescquote{%
\expand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
####3\expand\null}%
\expand\fi
\expand\fi
\expand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*\@gls@nosetquote}[1]{\PackageError{glossaries}%
{\string\GlsSetQuote\space not permitted here}%
{Move \string\GlsSetQuote\space earlier in the preamble, as
soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*\@gls@extramakeindexopts}[1]{
\newcommand{\noist}{%
\@gls@addpredefinedattributes
\let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
\GlossariesWarning{too late to use \string\noist. The style file
has already been written}%
}
\newcommand*\@makeglossary}[1]{%
\ifglossaryexists{#1}%
{%
\ifglssavewrites
\expandafter\newtoks\csname glo@#1@filetok\endcsname
\else
\expandafter\newwrite\csname glo@#1@file\endcsname
\expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
\fi
\@gls@renewglossary
\writeist
}%
}%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%

```

```

    }%
  }
  \newcommand*{\@glsopenfile}[2]{%
    \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
    \PackageInfo{glossaries}{Writing glossary file
      \jobname.\csname @glotype@#2@out\endcsname}%
  }
  \newcommand*{\@closegls}[1]{%
    \closeout\csname glo@#1@file\endcsname
  }
  \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@istfilename\space
        \@gls@codepage
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}}%
      }%
      \@gls@dothiswrite
    }%
  }
  \else
  \newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
      \@closegls{#1}%
      \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-l }}%
        {\let\@gls@order\@empty}%
      \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
        -s \istfilename\space
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}

```

```

        \jobname.\csuse{@glotype@#1@out}}%
    }%
    \@gls@dothiswrite
}%
{%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \gls@xdy@set@lang{#1}%
            \ifdefvoid\gls@codepage
            {\let\@gls@codepage\@empty}%
            {\def\@gls@codepage{-C \gls@codepage\space}}%
            \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
                -I xindy
                \@gls@order
                -L \@gls@langmod\space
                -M \gls@istfilebase\space
                \@gls@codepage
                -t \jobname.\csuse{@glotype@#1@log}
                -o \jobname.\csuse{@glotype@#1@in}
                \jobname.\csuse{@glotype@#1@out}}%
            }%
            \@gls@dothiswrite
        }%
        {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
            doesn't exist. Rerun may be required}}%
    }%
    {%
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-1 }}%
            {\let\@gls@order\@empty}%
        }
    }
}

```

```

\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@gls@glotype@#1@log}
-o \jobname.\csuse{@gls@glotype@#1@in}
\jobname.\csuse{@gls@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@gls@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@gls@types}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\else
\ifdefempty\gls@automake@makegloss{}%
{%
\IfFileExists{\jobname.aux}%
{\immediate\write18{\@gls@automake@makegloss\space \jobname}}{}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
}%
\fi
}
\newcommand*{\makeglossaries}{%
\ifglxindy
\def\glsindexingsetting{xindy}%
\else
\def\glsindexingsetting{makeindex}%
\fi
\@domakeglossaries
{%
\@gls@automake@immediate
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }

```

```

\ifundef\@gls@extramakeindexopts
{}%
{%
  \protected@write\@auxout{}\string\providecommand
    \string\@gls@extramakeindexopts[1]{%
  \protected@write\@auxout{}\string\@gls@extramakeindexopts
    {\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}\string\@glsorder{\glsorder}%
\@for\@glo@type:=\@glo@types\do{%
  \ifthenelse{\equal{\@glo@type}{}}{}}{%
  \makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][{}]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglsavenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
  \renewcommand*\@gls@doautomake}{%
    \protected@edef\gls@tmp{\@gls@automake@types}%
    \@for\@gls@type:=\gls@tmp\do{%
      \ifdefempty{\@gls@type}{%
        {\@gls@automake{\@gls@type}}%
      }%
    }%
  }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%

```

```

    \GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
    \GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
\AtEndDocument{%
    \warn@nomakeglossaries
    \warn@noprintglossary
}
\newcommand*\makenoidxglossaries{%
\def\glsindexingsetting{noidx}%
\@domakeglossaries
{%
    \renewcommand{\@gls@nofwarn}[1]{%
        \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
        \protected@edef\@gls@label{\glsdetoklabel{##1}}%
        \protected@write\@auxout{}{%
            \string\@gls@reference
            {\csname glo@\@gls@label @type\endcsname}%
            {\@gls@label}%
            {%
                \string\glsseeformat##2}%
            }%
        }%
    }%
}%
\AtBeginDocument
{%
    \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
    \GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%

```

```

\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{}%
    \def\@gls@noidxloclist@prev{}%
    \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
    \@gls@noidxloclist@finalsep
    \@gls@noidxloclist@prev
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}

```

```

    }%
  }%
}%
\renewcommand*\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%

```

```

\string\@gls@reference
  {\csname glo@\@gls@label @type\endcsname}%
  {\@gls@label}%
  {\string\glsnoidxdisplayloc
    {\@glo@counterprefix}%
    {\@gls@counter}%
    {\@glsnumberformat}%
    {\@glslocref}}%
  }%
}%
}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%
        \expandafter\the
          \csname glo@\@glo@type @filetok\endcsname}%
      \immediate\closeout\glswrite
    \fi
  }%
}
\newcommand*{\gls@glossary}[1]{%
  \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}

```

```

}
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
    {%
      \expandafter\protected@write\csname glo@#1@file\endcsname{%
        \gls@disablepagerefexpansion}{#2}%
      }%
    {%
      \ifignoredglossary{#1}{}%
      {%
        \GlossariesWarning{No file defined for glossary '#1'}%
      }%
    }%
  \fi
  \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\gls@writeentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*{\gls@writeentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
  \gls@wrglossdisablelocationcmds
  \ifglsclocations
    \@for\@gls@this:=\gls@protected@pagefmts\do
    {%
      \expandafter\let\@gls@this\relax
    }%
  \fi
}
\newcommand{\gls@wrglossdisablelocationcmds}{}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}

```

```

\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gls#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}
\newcommand*\GlsWarnAddProtectedPageFmt){%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand*\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*\@do@wrglossary}[1]{%
  \ifglsclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand*\glswrglossdisableanchorcmds}{\let\gls@texorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand*\glswrglosslocationtarget}[1]{#1}
\newcommand*\glswrglosslocationtextfmt}[1]{#1}
\newcommand*\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@gls@locref\expandafter{\the@gl@entrycounter}%
  \expandafter\def\expandafter\@gls@Hlocref\expandafter{\the@Hgl@entrycounter}%
  \ifx\@gls@Hlocref\@gls@locref
    \def\@gls@counterprefix{}%
  \else
    \@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
  \fi
}

```

```

\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\@xp@gls@getcounterprefix{\@gls@locref}{\@gls@Hlocref}%
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newcommand*{\@do@wrglossary}{%
\ifglswrallowprimitivemods
\expandafter\@glo@check@mkidxrangechar\@gls@numberformat\@nil

```

```

\def\@glo@range{}%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string"\@glo@counterprefix}{\@gls@locref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string@glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
\fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bggroup
\gls@wrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%
\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\@do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you

```

```

        will get the warning: "'name{\@gls@counter.#1}' has been^^J
        referenced but does not exist"%
    \fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:key (\csname glo@#1@index\endcsname)
:xref (\string"\@gls@xref\string")
:attr \string"see\string"
)
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string@glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{#{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*{\glssee}[3][\seename]{%
\@do@seeglossary{#2}{[#1]{#3}}
\newcommand*{\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*{\glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}
\newrobustcmd*{\glsseelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext
\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseelastsep

```

```

\let\@gls@donext\glsseesep
}%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat[1]{%
\ifgls hasshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
\newcommand*\gls@save@numberlist[1]{%
\ifgls@savenumberlist
\toks@{#1}%
\protected@edef\@do@writeaux@info{%
\noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\@do@writeaux@info
\protected@write\@auxout{}{\@do@writeaux@info}%
\fi
}
\newcommand*\warn@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}{%
  \foralllglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort}{%
\newcommand*\@gls@preglossaryhook}{%
\newcommand*\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary '\@glo@type'. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\glssettoc@title{\@glo@type}}%
  \let\@org@glossaryentrynumbers\glossaryentrynumbers
  \bgroup
  \@printgloss@setsort
  \setkeys{printgloss}{#1}%
  \@printgloss@checkexists{\@glo@type}%
  {%
    \ifx\glossarytitle\org@glossarytitle
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary\@glo@type
    \let\org@glossaryentrynumbers\glossaryentrynumbers
    \let\glsnonextpages\@glsnonextpages
    \let\glsnextpages\@glsnextpages
    \let\nopostdesc\@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
  }%
}

```

```

    \@gls@preglossaryhook
    #2%
  }%
\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}%
  \ifglsxindy
    \gls@xdy@set@lang{\@glo@type}%
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \noexpand\immediate\noexpand\write\@auxout{%
          \string\providecommand\string\@xdy@language[2]{}%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdy@language{\@glo@type}{\@gls@langmod}}%
        }%
      }%
    }%
    \@do@auxoutstuff
    \edef\@do@auxoutstuff{%
      \noexpand\AtEndDocument{%
        \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{\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{}%
\for@gl@sentries[#1]{\@gl@thislabel}%
{%
\ifinlistcs{\@gl@thislabel}{\@gl@ref@#1}%
{%
\listcsadd{\@glo@sortinglist}{\@gl@thislabel}%
}%
}%
}%
\cslet{\@gl@ref@#1}{\@glo@sortinglist}%
}
\newcommand*\@glo@sortmacro@def@do}[1]{%
\ifinlistcs{#1}{\@gl@ref@\@glo@type}%
{}%
{%
\listcsadd{\@gl@ref@\@glo@type}{#1}%
}%
\ifcsdef{\@glo@sortingchildren@#1}%
{%
\@glo@addchildren{\@glo@type}{#1}%
}%
{}%
}
\newcommand*\@glo@sortmacro@use}[1]{%
\newcommand*\@print@noidx@glossary}{%
\ifcsdef{\@gl@ref@\@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\@gl@currentlettergroup{}%
\begin{theglossary}%
\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gl@noidx@do}{\@gl@ref@\@glo@type}%
\end{theglossary}%
\glossarypostamble
}%

```

```

    {%
      \@gls@noref@warn{\@glo@type}%
    }%
  }
\def\glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%
  \ifdefempty\@glo@thislettergrp
  {%
    \def\@glo@thislettergrp{glssymbols}%
  }%
  {%
    \count@=\uccode‘#1\relax
    \ifnum\count@=0\relax
      \def\@glo@thislettergrp{glssymbols}%
    \else
      \ifdefstring\@glo@sorttype{case}%
      {%
        \count@=‘#1\relax
      }%
      {%
      }%
      \edef\@glo@thislettergrp{\the\count@}%
    \fi
  }%
}
\newcommand{\@gls@noidx@do}[1]{%
  \global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
  \ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
    {%
      \subglossentry{\gls@level}{#1}{}%
    }%
    {%
      \subglossentry{\gls@level}{#1}%
    }%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}

```

```

    }%
  }%
  {%
    \letcs{\@gls@sort}{glo@glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {}%
    {%
      \ifdefempty{\@gls@currentlettergroup}{}%
      {%
        \global\let\@glo@thislettergrp\@glo@thislettergrp
        \glsgroupskip
      }%
      \glsgroupheading{\@glo@thislettergrp}%
    }%
    \global\let\@gls@currentlettergroup\@glo@thislettergrp
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glsnoidxprenumberlist{#1}%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%

```

```

        \@gls@noidxloclist@sep
        \@gls@noidxloclist@prev
        \def\@gls@noidxloclist@prev{#1}%
    }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
    \setentrycounter[#1]{#2}%
    \csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
    \glsdoifexistsorwarn{#2}%
    {%
        \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{%
            \ifinlistcs{#2}{\@glsref@#1}%
            {}%
            {\listcsgadd{\@glsref@#1}{#2}}%
            \ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
            {\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
            {}%
            \listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
        }%
    }
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
    \def\glossarytitle{#1}%
    \let\gls@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{\setglossarystyle{#1}}%
    }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
    \ifcase\@gls@numberedsection@nr\relax
        \renewcommand*\@@glossarysecstar{*}%
        \renewcommand*\@@glossaryseclabel{}%
    \or
        \renewcommand*\@@glossarysecstar{}%
    \endcase
}

```

```

\renewcommand*\@@glossaryseclabel}{}%
\or
\renewcommand*\@@glossarysecstar}{}%
\renewcommand*\@@glossaryseclabel{\label{glsautoprefix\@glo@type}}%
\or
\renewcommand*\@@glossarysecstar}{*}%
\renewcommand*\@@glossaryseclabel{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*\GlsEntryCounterLabelPrefix{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*\@glo@no@assign@sortkey[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*\@glo@assign@sortkey[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*\@glsnonextpages{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*\@glsnextpages{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}
\newcommand*\glsresetentrylist{%

```

```

\global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}
\newcommand*{\glsentryitem}[1]{%

```

```

\ifglentrycounter
  \glstepentry{#1}\glentrycounterlabel
\else
  \glresetsubentrycounter
\fi
}
\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glstepsubentry{#1}\glssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheglossdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@glstetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glssentrydesc{#1}%
  }%
}
\newcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*{\glossentrysymbol}[1]{%

```

```

\glsdoifexistsorwarn{#1}%
{%
  \glsentrysymbol{#1}%
}%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*{\glsgroupskip}{-}
\newcommand*{\glsgroupheading}[1]{}
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
  \def#2{#1}%
  \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{-}{\letcs#2{#1groupname}} }
}
\ExplSyntaxOff
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][ ]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
}

```

```

\fi
\def\glentrycounter{#2}%
}
\newcommand*\setglossarystyle}[1]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
}%
{%
\csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
\protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand{\newglossarystyle}[2]{%
\ifcsundef{@glsstyle@#1}%
{%
\expandafter\def\csname @glsstyle@#1\endcsname{#2}%
}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
}%
}
\newcommand{\renewglossarystyle}[2]{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
}%
{%
\csdef{@glsstyle@#1}{#2}%
}%
}
\newcommand*\glshnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
\def\glshypernumber#1{#1}%
}%
{%
\def\glshypernumber#1{\@xp@glshypernumber{#1}}
}
\newrobustcmd{\@xp@glshypernumber}[1]{%
\@glshypernumber#1\nohyperpage{}\@nil}
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
\ifx\#1\%
\else
\gls@delimR#1\delimR\delimR\%
\fi
\ifx\#2\%

```

```

\else
  #2%
\fi
\ifx\#3\%
\else
  \@glshypernumber#3\@nil
\fi
}
\def\gls@delimR#1\delimR #2\delimR #3\{%
\ifx\#2\%
  \gls@delimN{#1}%
\else
  \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
  \@gls@numberlink{#1}%
\else
  \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
  \else
  \bgroup
  \gls@rglossdisableanchorcmds
  \protected@edef\@gls@tmp{\gls@entrycounter\@gls@counterprefix
  \gls@rglosslocationtarget{\@gls@x}}%
  \@onelevel@sanitize\@gls@tmp
  \edef\@gls@x{\noexpand\egroup
  \noexpand\hyperlink{\@gls@tmp}%
  {\noexpand\gls@rglosslocationtextfmt{\the\toks@}}%
  }%
  \@gls@x
  \fi
\else
  \@gls@ReturnAfterFi{%
  \@gls@removespaces#2\@nil
  }%
}

```

```

\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }%
}
\newcommand{\newacronym}[4][ ]{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
  \renewcommand{\gls@newacronym}[4]{%
    \PackageError{glossaries}{incompatible acronym style:
      rollback may be required}%
    {something has gone wrong with \string\newacronym. Try
      \string\usepackage{glossaries}[=v4.49]}%
  }%
  \setacronymstyle{long-short}%
  \newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
\newrobustcmd*\glsstextup[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey{short}
\newcommand*\glsshortpluralkey{shortplural}
\newcommand*\glslongkey{long}
\newcommand*\glslongpluralkey{longplural}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[ ]}%

```

```

}
\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][{}]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][{}]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}

```

```

}
\def\acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acrenryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
    }%
  }%
}

```

```

text={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\glspluralsuffix},%
\GenericAcronymFields,%
\the\glskeylisttok
}%
}%
\do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1][##2]{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1][##2]{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1][##2]{%
\glsupercase{\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]{%
\glsupercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*{\acronymsort}[2]{##1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@##1}
{%
\PackageError{glossaries}{Undefined acronym style ‘##1’}{%
}%
}%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{##1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{##1}}%
}%
}%

```

```

}
\newcommand*\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
\newcommand*\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
  }%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
  \renewcommand*\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*\acronymsort}[2]{##1}%
  \renewcommand*\acronymfont}[1]{##1}%
  \renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}

```

```

}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\protect\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
\glsentrylong{##1}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
\glsentrylong{##1}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
\glsentrylongpl{##1}%
}

```

```

}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})%
  }
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%

```

```

    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
}
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
}
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})%
  }
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \glsupercase
          {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%

```

```

        \Glsentrylong{\glslabel}\glsinsert
    }%
    {%
        \glsuppercase
        {\glsentrylong{\glslabel}\glsinsert}%
    }%
} %
} %
{%
    \glsgenentryfmt
} %
} %
{\glscustomtext\glsinsert}%
} %
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsuppercase{\glsentrylong{##2}##3\space
            (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
        (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
        (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsuppercase{\glsentrylongpl{##2}##3\space
            (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*{\glsentryfull}[1]{%
    \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
} %
\renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
} %
\renewcommand*{\glsentryfullpl}[1]{%
    \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
} %
\renewcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
} %
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%

```

```

\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
\ifglsashaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}}%
}

```

```

\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}}}%
\renewcommand*\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*\acrpluralsuffix{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\{GenericAcronymFields}\{}%
  \renewcommand*\{acronymsort}[2]{##2}%
  \renewcommand*\{acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}

```

```

\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \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
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{##1}{##2}{##3}}
\newcommand*{\acrlinkfootnote}[3]{%
  \footnote{\glslink{##1}{##2}{##3}}%
}

```

```

\newcommand*\acrnolinkfootnote}[3]{%
  \footnote{#3}%
}
\newrobustcmd*\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
\newcommand*\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
  Use \string\setacronymstyle{#2} with just glossaries.sty or
  use \string\setabbreviationstyle[acronym]{#3} with
  glossaries-extra.sty}%
}
\RequirePackage{glossary-hypernav}[=v4.52]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}

```

```

\newcommand*\showglocounter}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@counter\endcsname
}
\newcommand*\showglouserii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userii\endcsname
}
\newcommand*\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriii\endcsname
}
\newcommand*\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\endcsname
}
\newcommand*\showglouserv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userv\endcsname
}
\newcommand*\showglouservi}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@uservi\endcsname
}
\newcommand*\showglongname}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@name\endcsname
}
\newcommand*\showglodesc}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
}
\newcommand*\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
}
\newcommand*\showglosort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
}
\newcommand*\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
}
\newcommand*\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*\showgloshort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
}
\newcommand*\showglolong}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@long\endcsname
}
\newcommand*\showgloindex}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@index\endcsname
}
\newcommand*\showgloflag}[1]{%
  \expandafter\show\csname ifglo@glstdetoklabel{#1}@flag\endcsname
}

```

```

}
\newcommand*\showgloclist}[1]{%
  \expandafter\show\csname glo@glstoklabel{#1}@loclist\endcsname
}
\newcommand*\showglofield}[2]{%
  \csshow{glo@glstoklabel{#1}@#2}%
}
\newcommand*\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*\showglossaries}{%
  \show\@glo@types
}
\newcommand*\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glo@list@#1\endcsname
}
}

```

8.36 Rollback v4.52 (glossary-hypernav-2022-11-03.sty)

Version 4.52 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-hypernav}[2022/11/03 v4.52 (NLCT)]
\newcommand*\glsnavhyperlink}[3][\currentglossary]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
\newcommand*\glsnavhyperlinkname}[2]{glsn:#1@#2}
\newcommand*\glsnavhypertarget}{\protect\@gls@navhypertarget}
\newcommand*\@gls@navhypertarget}[3][\currentglossary]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*\@glsnavhypertarget}[3]{%
  \protected@write\@auxout}{\string\@gls@hypergroup{#1}{#2}}%
  \@gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
  \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }%
}

```

```

\if@endfor
\else
  \GlossariesWarningNoLine{Navigation panel
    for glossary type '#1'^^Jmissing group '#2'}%
  \gdef\gls@hypergroup\prerun{%
    \GlossariesWarningNoLine{Navigation panel
      has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup\relax
\AtEndDocument{\gls@hypergroup}
\newcommand*{\gls@hypergroup}[2]{%
  \ifcsdef{@gls@hypergroup@list@#1}%
  {\protected@csxappto{@gls@hypergroup@list@#1}{, #2}}%
  {\protected@csxdef{@gls@hypergroup@list@#1}{#2}}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{@gls@hypergroup@list@\currentglossary}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \letcs\@gls@list{@gls@hypergroup@list@\currentglossary}%
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
    \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
    \expandafter\glsnavhyperlink\expandafter{\@gls@tmp}{\@gls@grptitle}%
    \let\@gls@between\glsnavsep
  }%
}
\newcommand*{\glsnavsep}{\space\textbar\space}
\newcommand*{\glsymbolnav}{%
  \glsnavhyperlink{glsymbols}{\glsgetgrouptitle{glsymbols}}%
  \glsnavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glsnavsep
}

```

9 Rollback

9.1 Rollback v4.54 (glossaries-2024-04-03.sty)

Version 4.54 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries}[2024/04/03 v4.54 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]

```

```

\RequirePackage{mfirstuc}[=2.08]
\newcommand{\glssentencecase}[1]{\makefirstuc{#1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{ \text_lowercase:n { #1 } }
\newcommand{\glsuppercase}[1]{ \text_uppercase:n { #1 } }
\ifdef\MFUsentencecase
{%
  \newcommand{\glsmfuexcl}[1]{\MFUexcl{#1}}
  \newcommand{\glsmfublocker}[1]{\MFUblocker{#1}}
  \newcommand{\glsmfuaddmap}[2]{\MFUaddmap{#1}{#2}}
}
{
  \RequirePackage{textcase}
  \providecommand\MFUsentencecase[1]{ \text_titlecase_first:n { #1 } }
  \providecommand{\glsmfuexcl}[1]{
    \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
    {
      \tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
    }
  }
  \renewcommand*\mfirstucMakeUppercase{\MakeTextUppercase}
  \newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
  \newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}[=2.32]
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
{%
  \@gls@docloadedtrue
}%
{%
  \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug

```

```

\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \let\@glsshowtarget\@gobble
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
  \or
    % debug=false
    \@gls@debugfalse
    \let\@glsshowtarget\@gobble
    \let\@glsshowaccsupp\@gobblethree
    \PackageInfo{glossaries}{debug mode OFF}%
  \or
    % debug=showtargets
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \renewcommand{\@glsshowtarget}{\@glsshowtarget}%
  \or
    % debug=showaccsupp
    \@gls@debugtrue
    \renewcommand*\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
    \renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
  \fi
}
\newcommand*\glsshowtarget}[1]{%
  \ifmmode
    \glsshowtargetinner{#1}%
  \else
    \ifinner
      \glsshowtargetinner{#1}%
    \else

```

```

        \glsshowtargetouter{#1}%
    \fi
\fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
  \ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
  \glsshowtargetsymbolsymbol\marginpar{\glsshowtargetsymbolsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbolsymbol{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\glsshowtarget}[1]{%
  \newcommand*\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*\@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*\@glsshowaccsupp}[3]{%
\newrobustcmd*\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
  {You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x}{error,warn,ignore}{%
  [\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*\gls@seeautonumberlist{}
\gls@declareoption{seeautonumberlist}{%
\renewcommand*\gls@seeautonumberlist}{%
\def\gls@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationfalse
\newcommand*\gls@loadlong{\RequirePackage{glossary-long}}
\gls@declareoption{noalong}{\renewcommand*\gls@loadlong{}}
\IfFileExists{supertabular.sty}{%
\newcommand*\gls@loadsuper{\RequirePackage{glossary-super}}}{%
\newcommand*\gls@loadsuper{}}
\ifpackageloaded{xtab}{\renewcommand*\gls@loadsuper{}}{}}
\gls@declareoption{nosuper}{\renewcommand*\gls@loadsuper{}}
\newcommand*\gls@loadlist{\RequirePackage{glossary-list}}
\gls@declareoption{noindex}{%
\renewcommand*\gls@loadlist}{%
\ifdefstring{\gls@default@style}{list}%
{\let\gls@default@style\relax}%
}%
}%
}
\newcommand*\gls@loadtree{\RequirePackage{glossary-tree}}
\gls@declareoption{notree}{\renewcommand*\gls@loadtree{}}
\gls@declareoption{nostyles}{%
\renewcommand*\gls@loadlong}{%
\renewcommand*\gls@loadsuper}{%
\renewcommand*\gls@loadlist}{%
\renewcommand*\gls@loadtree}{%
\let\gls@default@style\relax
}
}
\newcommand*\gls@postdescription}{%
\ifglsnopostdot\else.\spacefactor\sfcode`\.\ \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\gls@nopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{%
\gls@nogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{%
\ifclassloaded{memoir}

```

```

{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
    \ifundef\c@glossaryentry
      {%
        \ifx\@gls@counterwithin\@empty
          \newcounter{glossaryentry}%
        \else
          \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
      }%
    }%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glsentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glsentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
    {%
      \ifglssubentrycounter
        \ifglsentrycounter
          \newcounter{glossarysubentry}[glossaryentry]%
        \else
          \newcounter{glossarysubentry}%
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
      \fi
    }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglsxindy xindy\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}

```

```

\newcommand*\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*\@glo@check@sortallowed}[1]{%
\newcommand*\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@glsdefaultsort
      \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort
    \glsprestandardsort{\@glo@sort}{##1}{##2}%
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
\newcommand*\@gls@setupsort@def}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
  }%
  \def\@gls@defsort##1##2{%
    \ifcsundef{glossary@##1@sortcount}%
    {\@gls@defsortcount{##1}}%
    }%
    \expandafter\global\expandafter
    \advance\csname glossary@##1@sortcount\endcsname by 1\relax
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
      \expandafter\glsortnumberfmt
      {\csname glossary@##1@sortcount\endcsname}}%
  }%
  \def\@gls@setsort##1{%
  \let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use}{%
  \let\do@glo@storeentry\@gobble
  \def\@gls@defsortcount##1{%
    \expandafter\global

```

```

\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
}
\newcommand*{\@gls@setupsort@clear}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=clear not allowed with \string##1}%
{(Use sort=def instead)}}%
}
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else

```

```

        \newglossary{main}{gls}{glo}{\glossaryname}%
    \fi
    \newcommand*\gls@tr@set@main@toctitle}{%
        \translatelet{\glossarytoctitle}{Glossary}%
    }%
}
\newcommand*\glsdefaulttype{main}
\newcommand*\acronymtype{\glsdefaulttype}
\@gls@declareoption{nomain}{%
    \let\glsdefaulttype\relax
    \renewcommand*\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
    \ifglsacronym
        \renewcommand*\@gls@do@acronymsdef{%
            \DeclareAcronymList{acronym}%
            \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
            \renewcommand*\acronymtype{acronym}%
            \newcommand*\gls@tr@set@acronym@toctitle}{%
                \translatelet{\glossarytoctitle}{Acronyms}%
            }%
        }%
    \else
        \let\@gls@do@acronymsdef\relax
    \fi
}
\AtBeginDocument{%
    \ifglsacronym
        \providecommand*\printacronyms[1][1]{%
            \printglossary[type=\acronymtype,#1]%
        }%
    \fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
    \glsacronymtrue
    \def\@gls@do@acronymsdef{%
        \DeclareAcronymList{acronym}%
        \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
        \renewcommand*\acronymtype{acronym}%
        \newcommand*\gls@tr@set@acronym@toctitle}{%
            \translatelet{\glossarytoctitle}{Acronyms}%
        }%
    }%
}
\newcommand*\@glsacronymlists{}
\newcommand*\@addtoacronymlists[1]{%
    \ifx\@glsacronymlists\@empty
        \protected@xdef\@glsacronymlists{#1}%
    \else
        \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
    \fi
}

```

```

\fi
}
\newcommand*\DeclareAcronymList}[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists}[1]{%
  \renewcommand*\@glsacronymlists{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*\glscounter{#1}%
}
\newcommand*\@gls@nohyperlist{}
\newcommand*\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*\@gls@nohyperlist{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}

```

```

}
\newcommand*\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*\GlossariesWarning[1]{}%
    \renewcommand*\GlossariesWarningNoLine[1]{}%
    \renewcommand*\glosortentrieswarning{}%
    \renewcommand*\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheglossdefined}{%
  \GlossariesWarning{Overriding 'theglossary' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*\@gls@warnonglossdefined}{}%
  \renewcommand*\@gls@warnontheglossdefined}{}%
}
\newcommand*\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}

```

```

\glsetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{ }
\glsetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{ }
\glsetnoexpandfield{symbol}
\glsetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifglssanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*\@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{ }
\newcommand*\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
    { }%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
    { }%
  {%
    \expandafter\@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x@\@secondoftwo
  \let\@tabacckludge@\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
}

```

```

\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let'\@firstofone
\let'\@firstofone
\let^\@firstofone
\let"@@firstofone
\let\u@firstofone
\let\t@firstofone
\let\d@firstofone
\let\r@firstofone
\let=\@firstofone
\let.\@firstofone
\let~\@firstofone
\let\v@firstofone
\let\H@firstofone
\let\c@firstofone
\let\b@firstofone
\let\a@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else

```

```

        \glsssetexpandfield{desc}%
        \glsssetexpandfield{descplural}%
    \fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
    \GlossariesWarning{sanitize={name} package option deprecated}%
    \ifglS@sanitize@name
        \glsssetnoexpandfield{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
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
    \ifglSsanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@glS@noidx@setsanitizesort}{%
            \glSsanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@glS@noidx@setsanitizesort}{%
            \glSsanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glSsanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@glS@noidx@setsanitizesort}{%
    \glSsanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
    \setbool{glSsanitizesort}{#1}%
    \ifglSsanitizesort
        \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}%
      \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{\glsifusedtranslator} [3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}{%
  [\gls@translate@val\gls@translate@nr]%
  {true,false,babel}[true]%
  {%

```

```

\ifcase\gls@translate@nr\relax
\glstranslatetrue
\renewcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
}
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
  \glstranslatetrue
  \let\glsifusetranslator\@firstoftwo
}%
{%
  \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
  {
    \@ifpackageloaded{\gls@thissty}%
    {%
      \glstranslatetrue
      \@endfortrue
    }%
  }%
}
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%

```

```

\renewcommand{\@gls@requested@languages}{#1}%
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*\@gls@setacrstyle{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifglsacrfootnote
  \@gls@removedoption{footnote}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
\ifglsacrdescription
  \@gls@removedoption{description}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}
\ifglsacrsmallcaps
  \@gls@removedoption{smallcaps}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{}
\ifglsacrsmaller
  \@gls@removedoption{smaller}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{}
\ifglsacrdua
  \@gls@removedoption{dua}%
\fi
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{}
\newcommand*\@glsorder}{word}
\newcommand*\@glsorder}[1]{}
\define@choicekey{glossaries.sty}{order}{word,letter}{}
  \def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{}
\gls@xindy@glsnumberstrue
\def\@xdy@main@language{\@languagename}%
\newcommand*\@gls@xdy@set@lang}[1]{}
\ifcsundef{\@xdy@#1@language}%
  {}
  \ifcsundef{\@xdy@\glsdefaulttype @language}

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy\@glsdefaulttype @language}}%
    }%
    {\letcs\@gls@langmod{\@xdy@#1@language}}%
}
\define@key[gls]{xindy}[language]{\def\@xdy@main@language{#1}}
\ifvoid\inputencodingname
  {\def\gls@codepage{utf8}}%
  {\def\gls@codepage{\inputencodingname}}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key[glossaries.sty]{xindy}[][%
  \glsxindytrue
  \setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
  \glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
  \glsxindytrue
  \gls@xindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{\if@files #1\fi}
\@gls@declareoption{disablemakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘disablemakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘disablemakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}}%
    }%
  }%
  \renewcommand{\@domakeglossaries}[1]{%
    \PackageInfo{glossaries}{\string\makeglossaries\space and
      \string\makenoidxglossaries\space have been disabled}%
  }%
  }%
}
\@gls@declareoption{restoremakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble

```

```

    {%
      \GlossariesWarning{Option '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
    }%
  }%
}
\newcommand*{\@do@write@glslabels}{%
  \AtEndDocument{\@do@write@glslabels}%
  \let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
  \newwrite\@gls@labelsfile
  \immediate\openout\@gls@labelsfile=\jobname.glslabels
  \protected@edef\@gls@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{%
    {\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
  }%
  \forallglsentries[\@gls@fulllist]{\@glsentry}%
  {%
    \ifdefempty{\@glsentry}{%
      {%
        \ifx\do@write@glslabelsname@gobble
          \def\@gls@thisentryname{%
        }%
        \else
          \letcs{\@gls@thisentryname}{glo@\@glsdetoklabel{\@glsentry}@name}%
          \@onelevel@sanitize\@gls@thisentryname
        \fi
        \immediate\write\@gls@labelsfile
          {\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
      }%
    }%
  }%
  \immediate\closeout\@gls@labelsfile
}
\newcommand*{\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@gls@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname@gobble}
  \@gls@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglsautomake
  \newcommand{\gls@automake@nr}{1}
  \newcommand{\gls@automake@makegloss}{%
  \newcommand{\gls@automake@option}{automake=\gls@automake@val}

```

```

\newcommand*\@gls@do@automake@err}{%
  \PackageError{glossaries}{You must use
  \string\makeglossaries\space with \gls@automake@option}
  {%
    Either remove the \gls@automake@option\space setting or
    add \string\makeglossaries\space to your document preamble.%
  }%
}%
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\gls@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{}%
  \or
    \glsautomakefalse
    \renewcommand{\gls@automake@makegloss}{}%
  \or
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{}%
  \or
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{makeglossaries}%
  \or
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
  \or
    \def\gls@automake@nr{0}%
    \glsautomaketrue
    \renewcommand{\gls@automake@makegloss}{}%
  \fi
  \renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
  \ifglsautomake
    \let\@gls@doautomake\@gls@do@automake@err
  \else
    \renewcommand*\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*\@gls@doautomake}{%
\AtEndDocument{\@gls@doautomake}
\@gls@declareoption{automakegloss}{%
  \glsautomaketrue
  \renewcommand{\gls@automake@option}{automakegloss}%
  \def\gls@automake@val{makegloss}%
  \def\gls@automake@nr{3}%
  \renewcommand{\gls@automake@makegloss}{makeglossaries}%
  \let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%

```

```

\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\@gls@doautomake\@gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\gls savewrites false
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
  \ifbool{glscompatible-3.07}%
  {%
    \@gls@removedoption{compatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \@gls@removedoption{compatible-2.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary@glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}%
\gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}{%
  [\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
}%
\ifcase\gls@kernelglossredefs@nr\relax
  \def\glossary{\@gls@override@glossary}%

```

```

\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}%
  \makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
  \gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glsymbolsgroupname}%
  \newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]}%
  \newcommand*{\gls@tr@set@symbols@toctitle}{%
    \translatelet{\glossarytoctitle}{Symbols (glossaries)}%
  }%
}%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]}%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
  }%
}%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
  \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}

```

```

\newcommand*{\@gls@do@indexdef}{-}
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\printglossary[type=index,##1]]%
  \newcommand*{\newterm}[2][\%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\define@choicekey{glossaries.sty}{mfirstuc}%
[\gls@mfirstuc@val\gls@mfirstuc@nr]%
{expanded,unexpanded}%
{%
  \ifcase\gls@mfirstuc@nr
  % expanded
  \renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
  \or
  % unexpanded
  \renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
  \fi
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \@expandtwoargs
  \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
  \ifin@
  \@use@ption
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
  \fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{-}
\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{-}%
  \ifglsacrshortcuts
  \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
  \def\@gls@setupshortcuts{%
    \ifglsacrshortcuts
    \DefineAcronymSynonyms
  }
  \fi
}%

```

```

\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifglsentrycounter
  \let\@gls@doentrycounterdef\relax
\else
  \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifglssubentrycounter
  \let\@gls@dosubentrycounterdef\relax
\else
  \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}%
{%
  \ifcsundef{chapter}{}%
  {%
    \let\@gls@old@chapter\@chapter
    \def\@chapter[#1]#2{\@gls@old@chapter[#{1}]{#2}}%
    \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}{}}%
  }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
  \ifx\@gls@onlypremakeg\@empty
    \def\@gls@onlypremakeg{#1}%
  \else
    \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
    \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
  \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
  \expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%

```

```

\def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*\glossaryname{Glossary}
\providecommand*\acronymname{Acronyms}
\newcommand*\glssettoctitle[1]{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*\entryname{Notation}
\providecommand*\descriptionname{Description}
\providecommand*\symbolname{Symbol}
\providecommand*\pagelistname{Page List}
\providecommand*\glsymbolsgroupname{Symbols}
\providecommand*\glsnumbersgroupname{Numbers}
\newcommand*\glspluralsuffix{s}
\newcommand*\glsacrpluralsuffix{\glspluralsuffix}
\newcommand*\glsupacrpluralsuffix{\glstextup{\glsacrpluralsuffix}}
\providecommand*\seename{see}
\providecommand*\andname{&}
\newcommand*\RequireGlossariesLang[1]{%
\@ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}}%
}
\newcommand*\ProvidesGlossariesLang[1]{%
\ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*\addglossarytocaptions[1]{}
\ifglstranslate
\RequirePackage{tracklang}[2019/11/30]
\ifdefvoid\@gls@requested@languages{}%
{%
\@for\@gls@thislang:=\@gls@requested@languages\do{%
\TrackIfKnownLanguage{\@gls@thislang}{}%
{\GlossariesWarning{unknown language '\@gls@thislang'}}%
}%
}%
\@gls@usetranslator
\@ifpackageloaded{translator}
{%
\ifboolexpr
{
test {\ifdefstring{\trans@languages}{English}}
and not
test {\ifdefstring{bbl@loaded}{english}}
}
{%
\let\glsifusetranslator\@secondoftwo
}%
{%
\usedictionary{glossaries-dictionary}%
\renewcommand*\addglossarytocaptions[1]{%

```

```

\ifcsundef{captions#1}{}%
{%
  \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
  \expandafter\toks@\expandafter{\@gls@tmp
    \renewcommand*{\glossaryname}{\translate{Glossary}}}%
  }%
  \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
}%
}%
}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
}

```

```

    }{}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*\@no@post@desc{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*\glstfilebase{#1}%
  \ifglxindy
    \def\istfilename{\glstfilebase.xdy}
  \else
    \def\istfilename{\glstfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\glstfilebase.xdy}
\else
  \def\istfilename{\glstfilebase.ist}
\fi
\newcommand*\glstfilebase{\jobname}
\newcommand*\@istfilename[1]{
\newcommand*\glscpositor}{.}
\newcommand*\glstSetCompositor}[1]{%
  \renewcommand*\glscpositor{#1}}
\@onlypremakeg\glstSetCompositor
\newcommand*\@glstAlphaCompositor{\glscpositor}
\ifglxindy
  \newcommand*\glstSetAlphaCompositor[1]{%
    \renewcommand*\@glstAlphaCompositor{#1}}
\else
  \newcommand*\glstSetAlphaCompositor[1]{%
    \glstnoindywarning\glstSetAlphaCompositor}
\fi
\@onlypremakeg\glstSetAlphaCompositor
\newcommand*\glstsuffixF{}
\newcommand*\glstSetSuffixF}[1]{%
  \renewcommand*\glstsuffixF{#1}}
\@onlypremakeg\glstSetSuffixF
\newcommand*\glstsuffixFF{}
\newcommand*\glstSetSuffixFF}[1]{%
  \renewcommand*\glstsuffixFF{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*\glstnumberformat}[1]{#1}%

```

```

}%
{%
  \newcommand*{\glsnumberformat}[1]{\glsnumber{#1}}%
}
\newcommand{\glslocationstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@pglossarysection{#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{\glsupercase{#1}}{\glsupercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}

```

```

}
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \mkboth{\glsupercase{#1}}{\glsupercase{#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@ifinlist{%
    \def\gls@listsuffix{##2}%
    \ifx\gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \do@ifinlist,#2,#1,\end@ifinlist
}
\ifglsxindy
  \newcommand*\xdycounters{\glscounter}
  \newcommand*\GlsAddXdyCounters[1]{%
    \@for\gls@ctr:=#1\do{%
      \edef\do@addcounter{%
        \noexpand\gls@ifinlist{\gls@ctr}{\xdycounters}{}%

```

```

        {%
        \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
        \noexpand\@gls@ctr}%
        }%
    }%
    \@do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnoxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glsaddxdycounters{%
\PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glsaddxdyattribute[2]{%
\edef\@xdyattributes{\@xdyattributes ^^J
\string"#2#1\string"}%
\expandafter\toks@\expandafter{\@xdylocref}%
\edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
\ifx\@xdyattributelist\@empty
\edef\@xdyattributelist{#1}%
\else
\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi
\edef\@xdyattributes{\@xdyattributes ^^J
\string"#1\string"}%
\@for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxdyattribute{%
\noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
}
\gls@do@addxdyattribute
}%
}

```

```

\let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxywarning\GlsAddXdyAttribute}
\fi
\ifglxindy
\newcommand*{\@gls@addpredefinedattributes}{%
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
\let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ^^J
(define-alphabet "#1" (#2))}}
\else
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
\newcommand*{\@gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%

```

```

arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@gls@alphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}

```

```

\fi
\ifglxindy
  \newcommand*\GlsAddXdyLocation}[3][ ]{%
    \def\@gls@tmp{#1}%
    \ifx\@gls@tmp\@empty
      \edef\@xdyuserlocationdefs{%
        \@xdyuserlocationdefs ^^J%
        (define-location-class \string"#2\string"^^J\space\space
          \space(:sep \string"{}\glsopenbrace\string" #3
            :sep \string"\glsclosebrace\string"))
      }%
    \else
      \edef\@xdyuserlocationdefs{%
        \@xdyuserlocationdefs ^^J%
        (define-location-class \string"#2\string"^^J\space\space
          \space(:sep "\glsopenbrace"
            #1
            :sep "\glsclosebrace\glsopenbrace" #3
            :sep "\glsclosebrace"))
      }%
    \fi
    \edef\@xdyuserlocationnames{%
      \@xdyuserlocationnames^^J\space\space\space
      \string"#2\string"}%
  }
  \@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation}[2]{%
    \glsnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnoxindywarning\GlsSetXdyLocationClassOrder}
\fi

```

```

\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
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][ ]{%
    \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
\AtBeginDocument{%
  \ifx\@gls@codepage\@empty

```

```

        \@ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
    \fi
}
\else
    \newcommand*\GlsSetXdyCodePage[1]{%
        \glsnoxindywarning\GlsSetXdyCodePage}
\fi
\ifglsxindy
    \ifglsexindy@glsnumbers
        \def\@xdylettergroups{(define-letter-group
            \string"glsnumbers\string"^^J\space\space\space
            :prefixes (\string"0\string" \string"1\string"
                \string"2\string" \string"3\string" \string"4\string"
                \string"5\string" \string"6\string" \string"7\string"
                \string"8\string" \string"9\string")^^J\space\space\space
            \@xdynumbergrouporder)}
    \else
        \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
    \expandafter\toks@\expandafter{\@xdylettergroups}%
    \protected@edef\@xdylettergroups{\the\toks@^^J%
        (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
    }%
\newcommand*\forallglossaries[3][\@glo@types]{%
    \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
    \@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}

\newcommand*\forglsentries[3][\glsdefaulttype]{%
    \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
    \@for#2:=\@glo@list\do
    {%
        \ifdefempty{#2}{-}{#3}%
    }%
}

\newcommand*\forallglsentries[3][\@glo@types]{%
    \expandafter\forallglossaries\expandafter[#1]{\@thisglo@}%
    {%
        \forglsentries[\@thisglo@]{#2}{#3}%
    }%
}

\newcommand\ifglossaryexists{%
    \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

\newcommand\@ifglossaryexists[3]{%
    \ifcsundef{glo@type@#1@out}{#3}{#2}%
}

```

```

}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
}
\newcommand{\glsdoifexists}[2]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@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
}%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcsemtyp{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsemtyp{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{@no@post@desc}%
  {#2}%
  {#3}%
}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}
}

```

```

\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glstdetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdequal\@glo@short\@gls@default@value
      {#3}%
    {#2}%
  }%
}
\newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@glstdetoklabel{#2}@#1}{#3}{#4}}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glstdetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        {%
          \ifdequal\@glo@thisvalue\@gls@default@value
            {%
              #4%
            }%
          {%
            \let\glscurrentfieldvalue\@glo@thisvalue
            #3%
          }%
        }%
      }%
    }%
  {%
    \@gls@fetchfield{\@gls@thisfield}{#1}%
    \ifdef\@gls@thisfield
      {%
        \letcs{\@glo@thisvalue}{glo@glstdetoklabel{#2}@#1}%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        {%
          \ifdequal\@glo@thisvalue\@gls@default@value
            {%
              #4%
            }%
          {%
            \let\glscurrentfieldvalue\@glo@thisvalue
            #3%
          }%
        }%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \GlossariesWarning{Unknown entry field ‘#1’}%
    #4%
  }%
}%
}%
}
\newcommand*\glscurrentfieldvalue{}
\newcommand*\glo@types}{,}
\newcommand*\gls@provide@newglossary{%
  \protected@write\auxout{}\string\providecommand\string@newglossary[4]{}%
  \let\gls@provide@newglossary\relax
}
\newcommand*\@gls@defaulttype{\glsdefaulttype}
\newcommand*\defglsentryfmt}[2][\glsdefaulttype]{%
  \protected@edef\glo@tmp{#1}%
  \ifx\glo@tmp\@gls@defaulttype
    \GlossariesWarning{\string\defglsentryfmt\space attempting to set
      a display style for \string\glsdefaulttype\space but there
      doesn’t seem to be a default glossary. If you have used
      ‘nomain’, you need to define an alternative glossary}%
  \else
    \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
  \fi
}
\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtext}[1]{%
  \ifboolexpr{test {\ifdefstring{#1}{tex}}
    or test {\ifdefstring{#1}{TEX}}}
  {%
    \def#1{nottex}%
    \PackageError{glossaries}%
      {Forbidden ‘.tex’ extension replaced with ‘.nottex’}%
      {I’m sorry, I can’t allow you to do something so reckless.\MessageBreak
        Don’t use ‘.tex’ as an extension for a temporary file.}%
  }%
  {%
  }%
}
\newcommand*\gls@gobbleopt{\new@ifnextchar[{\@gls@gobbleopt}]}
\def\@gls@gobbleopt[#1]{
\newcommand*\newglossary{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%

```

```

\ifundef\glsdefaultttype
{%
  \gdef\glsdefaultttype{#2}%
}%
\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{}%
\gls@gobbleopt
}%
}
\newcommand*\altnewglossary[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*\@newglossary[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*\@gls@getcounter[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*\newignoredglossary[1]{%
  \ifdefempty\@ignored@glossaries

```

```

{%
  \protected@edef\@ignored@glossaries{#1}%
}%
{%
  \protected@eappto\@ignored@glossaries{,#1}%
}%
\csgdef{glolist@#1}{,}%
\ifcsundef{gls@#1@entryfmt}%
{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}%
{}%
\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*\@gls@nohyperlist{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{,#1}%
}%
}
\newcommand*\@ignored@glossaries{}
\newcommand*\@ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*\@gls@default@value{\relax}

```

```

\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
}%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
\renewcommand*{\gls@xr@key}{#1}%
\gls@checkseeallowed
\def#2{#3}%
\@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
\@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
\GlossariesWarning
{‘\gls@xr@key’ key doesn’t have any effect when used in the document
environment. Move the definition to the preamble
after \string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}{%
[\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
\ifcase\gls@nonumberlist@nr\relax
\def\@glo@prefix{\glsnonextpages}%
\@gls@savenonumberlist{true}%
}
}

```

```

\else
  \def\@glo@prefix{\glsnextpages}%
  \@gls@savenonumberlist{false}%
\fi
}
\newcommand*\@gls@noidx@nonumberlist}[1]{%
  \cslet\glo@\@glo@label @prenumberlist\@glo@prefix
}%
\newcommand*\@gls@noidx@prenumberlist}[1]{%
  \cuse\glo@\@gls@detoklabel{#1}@prenumberlist}
\newcommand*\@gls@savenonumberlist}[1]{%
\newcommand*\@gls@initnonumberlist}{%
\newcommand*\@gls@storenonumberlist}[1]{%
\newcommand*\@gls@enablesavenonumberlist}{%
  \renewcommand*\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet\glo@\@gls@detoklabel{##1}@nonumberlist\@glo@nonumberlist}%
    }%
    {}%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
}

```

```

\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '@glo@label'}{You
  haven't specified the entry name}}
\newcommand*\@glsnodesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \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}%
\ifdequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\gls@expand@fields{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\gls@noexpand@fields{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%
{%
\gls@defglossaryentry{#1}{#2}%
}%
}
\newcommand*\gls@defdocnewglossaryentry{%
\let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
\ifglsentryexists{#1}%
{%
}%
\gls@defglossaryentry{#1}{#2}%
}%
}
\@onlypreamble{\provideglossaryentry}
\newcommand*\gls@glsdefs@labellist{}
\newcommand*\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*\gls@docdef@labellist{}
\newcommand*\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
\ifundef\@gls@deffile
{%
\global\newwrite\@gls@deffile
\immediate\openout\@gls@deffile=\jobname.glsdefs
}%
}%
\ifglsentryexists{#1}%
{%
\xifinlist{#1}{\gls@docdef@labellist}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in the
document environment (ignoring new definition)}%
}%
}%
\xifinlist{#1}{\gls@glsdefs@labellist}%
{%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
}

```

```

    \@gls@writedef{#1}%
  }%
  {%
    \GlossariesWarning{Entry ‘#1’ has already been defined in
      the preamble (in which case remove the duplicate definition)
      or you have just updated to glossaries.sty
      v4.47 (in which case delete ‘\jobname.glsdefs’)}%
  }%
}
}%
{%
  \gls@defglossaryentry{#1}{#2}%
  \gls@docdef@addlabel{#1}%
  \@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*{\gls@begindocdefs}{%
  \gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode‘\noexpand\@=\number\catcode‘\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{}{}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*{\glsdefs@newdocentry}[2]{%
  \ifglsentryexists{#1}%
  {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
    (if you have moved the definition there from the document
    environment, delete \jobname.glsdefs)}}%
  {%
    \gls@glsdefs@addlabel{#1}%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}}\@expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile

```

```

    {%
      \expandafter\@firstoftwo\@gls@map
      =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
      \glspercentchar
    }%
  }%
  {}%
}%
\gls.writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}

```

```

}%
}
\newcommand*\glsaddstoragekey{\@ifstar\sglsaddstoragekey\glsaddstoragekey}
\newcommand*\@sglsaddstoragekey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}%
}%
\@glsaddstoragekey{#1}%
}
\newcommand*\@glsaddstoragekey[3]{-%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  {%
    \PackageError{glossaries}{Key ‘#1’ already exists}{}%
  }%
}
\newcommand*\glsaddkey{\@ifstar\sglsaddkey\glsaddkey}
\newcommand*\@sglsaddkey[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*\@glsaddkey[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{\@glo@tmp}{@glo@#1}%
    }%
  }%
}

```

```

\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Glsentryfield{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  }%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @gls@user@#1@\endcsname}[2][ ]{%
  \new@ifnextchar[%
    {\csuse{@gls@user@#1@}{##1}{##2}}%
    {\csuse{@gls@user@#1@}{##1}{##2}[ ]}}%
\csdef{@gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
  }%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @Gls@user@#1@\endcsname}[2][ ]{%
  \new@ifnextchar[%
    {\csuse{@Gls@user@#1@}{##1}{##2}}%
    {\csuse{@Gls@user@#1@}{##1}{##2}[ ]}}%
\csdef{@Gls@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  }%
}%
{%

```

```

\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1\endcsname}[2][]{%
    \new@ifnextchar[%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{%
  \PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newrobustcmd{\glsfieldxdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\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}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%

```

```

    {%
      \expandafter\gdef\csname glo@\glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfielddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \expandafter\def\csname glo@\glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \letcs#3{glo@\glo@label @#2}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand*\glsunexpandedfieldvalue[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstring{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}

```

```

}%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\expandafter\ifdefstrequal\csname glo@\glo@label @#2\endcsname{#3}{#4}{#5}%
}%
}%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\glo@label @#2}%
{%
\ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
}%
}%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}
\newcommand*{\glswritedefhook}{}
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\@glo@desc}%
\gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bgroup
\let\@org@newglossaryentryprehook\@newglossaryentryprehook
\long\def\@newglossaryentryprehook{%
\long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
\@org@newglossaryentryprehook
}%
\renewcommand*{\gls@assign@desc}[1]{%
\global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
\global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
}
\gls@defglossaryentry{#1}{#2}%
\egroup
}
}
}

```

```

\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\@glo@label{\gls@detoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@gls@name
  \let\@glo@desc\@gls@desc
  \let\@glo@descplural\@gls@default@value
  \let\@glo@type\@gls@default@value
  \let\@glo@symbol\@gls@default@value
  \let\@glo@symbolplural\@gls@default@value
  \let\@glo@text\@gls@default@value
  \let\@glo@plural\@gls@default@value
  \let\@glo@first\@gls@default@value
  \let\@glo@firstplural\@gls@default@value
  \let\@glo@sort\@gls@default@value
  \let\@glo@counter\@gls@default@value
  \def\@glo@see{}%
  \def\@glo@parent{}%
  \def\@glo@prefix{}%
  \@gls@initnonumberlist
  \def\@glo@useri{}%
  \def\@glo@userii{}%
  \def\@glo@useriii{}%
  \def\@glo@useriv{}%
  \def\@glo@userv{}%
  \def\@glo@uservi{}%
  \def\@glo@short{}%
  \def\@glo@shortpl{}%
  \def\@glo@long{}%
  \def\@glo@longpl{}%
  \@newglossaryentryprehook
  \setkeys{glossentry}{#2}%
  \ifundef\glsdefaulttype
  {%
    \PackageError{glossaries}%
    {No default glossary type (have you used ‘nomain’ by mistake?)}%
    {If you use package option ‘nomain’ you must define
     a new glossary before you can define entries}%
  }%
  {}%
  \gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
  \protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
  \ifcsundef{glo@list@\@glo@type}%
  {%

```

```

\PackageError{glossaries}%
{Glossary type ‘\@glo@type’ has not been defined}%
{You need to define a new glossary type, before making entries
in it}%
}%
{%
\ifignoredglossary\@glo@type
{%
\ifx\@glo@desc\@glsnodesc
\let\@glo@desc\@empty
\fi
}%
{%
}%
\protected@edef\@glo@list@\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
\@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
\ifdefequal\@glo@label\@glo@parent%
{%
\PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
\def\@glo@parent{}%
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
{%
\ifglsentryexists{\@glo@parent}%
{%
\expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
\@glo@parent}%
\gls@level=\csname glo@\@glo@parent @level\endcsname\relax
\advance\gls@level by 1\relax
\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
\ifx\@glo@plural\@glsdefault@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
}%
\PackageError{glossaries}%
{%
Invalid parent ‘\@glo@parent’
for entry ‘\@glo@label’ - parent doesn’t exist%
}%

```

```

    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glo@type@\@glo@type @counter}%
  {%
    \def\@glo@defaultcounter{\glscounter}%
  }%
  {%
    \letcs\@glo@defaultcounter{@glo@type@\@glo@type @counter}%
  }%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@gls@name
  \gls@name
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%

```

```

\ifcsundef{glo@\@glo@label @numberlist}%
{%
  \csxdef{glo@\@glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@desc{\@glo@first}%
\ifx\@glo@desc\@glo@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@gls@nodesc
  \@gls@nodesc
  \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\ifftrue
}%
\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@glsee{%
    \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
    \noexpand\expandafter\noexpand\@glsee\noexpand\@glo@list{\@glo@label}}%
  \do@glsee
}%
\@glo@autoseehook
}%
\newcommand*\@glo@autoseehook{}
\newcommand*\@newglossaryentryprehook{}
\newcommand*\@newglossaryentryposthook{}
\newcommand*\@newglossaryentry@defcounters{}
\newcommand*\@glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \protected@edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\glo@list{,%}
  \for@glseentries[\glo@type]{\glo@label}%
  {%
    \ifdefequal\@glo@thislabel\glo@label
      {\protected@ea@pto\glo@list{\glo@label,}}%
    }%
  \cslet\glo@list@\glo@type{\glo@list}%
  \csdef\glo@\@glo@thislabel @type}{#2}%
}
\ifglxindy
  \newcommand*\@glossaryentryfield{\string\glossentry}
\else
  \newcommand*\@glossaryentryfield{\string\glossentry}
\fi
\ifglxindy
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\else
  \newcommand*\@glossarysubentryfield{%
    \string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \protected@edef\@glo@esclabel{#1}%
  \@gls@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gls@checkmkidxchars\@glo@sort
  \@gls@checkmkidxchars\@glo@prefix
  \protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \ifglxindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string\@glo@sort\string" %
          \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
        }%
    \else

```

```

\expandafter\protected\xdef\csname glo@#1@index\endcsname{%
  \csname glo@\@glo@parent @index\endcsname
  (\string"\@glo@sort\string" %
  \string"\@glo@prefix\@glossarysubentryfield
  {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
}%
\fi
\else
\ifx\@glo@parent\@empty
\@onelevel@sanitize\@glo@prefix
\expandafter\protected\xdef\csname glo@#1@index\endcsname{%
  \@glo@sort\@gls@actualchar\@glo@prefix
  \@glossaryentryfield{\@glo@esclabel}%
}%
\else
\expandafter\protected\xdef\csname glo@#1@index\endcsname{%
  \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
  \@glo@sort\@gls@actualchar\@glo@prefix
  \@glossarysubentryfield
  {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*\@gls@if@measuring}[2]{%
  \ifmeasuring@
  #1%
  \else
  #2%
  \fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand*\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}%
  {\let\gls@if@measuring\@gls@if@measuring}%
  {}%
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\gls@ifmeasuring{#1}}
\newcommand*\gls@ifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter}[1]{%
  \ifcsdef{c@#1}%
  {%
    \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
    \expandafter\advance\csname c@#1\endcsname by \@ne
  }%
  {}%
}

```

```

}
\newcommand{\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo
  \let\@glslink\@secondoftwo
}
\ExplSyntaxOn
\tl_if_exist:NT \l_datatool_measure_hook_tl
{
  \tl_put_right:Nn \l_datatool_measure_hook_tl
  { \gls@start@measuring }
}
\ExplSyntaxOff
\newcommand{\gls@stop@measuring}{%
  \renewcommand*{\glsifmeasuring}{\gls@if@measuring}%
  \let\refstepcounter\gls@org@refstepcounter
  \let\label\gls@org@label
  \let\glstarget\gls@org@target
  \let\@glslink\gls@org@link
  \gls@measuring@reset
  \def\gls@measuring@reset{}%
}
\newcommand{\gls@measureheight}[2]{%
  \settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\gls@measuredepth}[2]{%
  \settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\gls@measurewidth}[2]{%
  \settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\glspatchtabularx\relax
  }%
  {}%
}

```

```

\newcommand*\glsreset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsreset{#1}%
    }%
  }%
}
\newcommand*\glslocalreset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalreset{#1}%
    }%
  }%
}
\newcommand*\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%

```

```

\expandafter\global\csname glo@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}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*\glsenableentrycount){%
\let\@newglossaryentry@defcounters\@@newglossaryentry@defcounters
\renewcommand*\gls@defdocnewglossaryentry){%
\renewcommand*\newglossaryentry[2]{%
\PackageError{glossaries}{\string\newglossaryentry\space
may only be used in the preamble when entry counting has
been activated}{If you use \string\glsenableentrycount\space
you must place all entry definitions in the preamble not in
the document environment}%
}%
}%
\newcommand*\glsentrycurrcount}[1]{%
\ifcsundef{glo@glstetoklabel{##1}@currcount}%
{0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*\glsentryprevcount}[1]{%
\ifcsundef{glo@glstetoklabel{##1}@prevcount}%
{0}{\@gls@entry@field{##1}{prevcount}}%
}

```

```

}%
\renewcommand*\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*\@glsreset}[1]{%
  \@glsreset{##1}%
  \ifglsresetcurrcount\csgdef{glo@glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgl@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cglformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cgl@sp@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cgl@sp@format{##2}{##3}%
    \glsunset{##2}%
  \else
    \@gls@sp@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGlsformat{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@{##1}{##2}[##3]%
  \fi
}%
\def\@cGls@sp@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax
    \cGls@sp@format{##2}{##3}%
    \glsunset{##2}%
  \else
    \@Gls@sp@{##1}{##2}[##3]%
  \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%

```

```

\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{glo@glsdetoklabel{##1}@prevcount}{##2}%
}%
\let\glsenableentrycount\relax
}
\@onlypreamble\glsenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
  \csxdef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsdetoklabel{##1}@currcount}{%
    \number\numexpr\glsentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*{\string\@gls@entry@count}[2]{}}%
  \forallglsentries{\@glsentry}{%
    \ifglsused{\@glsentry}%
    {\immediate\write\@auxout
      {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}}%
    {}%
  }%
}
\newcommand*{\@gls@entry@count}[2]{%
}
\newrobustcmd*{\cglsl}{\@gls@hyp@opt\@cglsl}
\newcommand*{\@cglsl}[2][{}]{%
  \new@ifnextchar[{\@cglsl@{##1}{##2}}{\@cglsl@{##1}{##2} []]}%
}
\def\@cglsl@#1#2[#3]{%
  \GlossariesWarning{\string\cglsl\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@gls@{##1}{##2}[##3]}%
}
\newcommand*{\cglslformat}[2]{%
  \ifglshaslong{##1}{\glsentrylong{##1}}{\glsentryfirst{##1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\glsmfuaddmap{\cglsl}{\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*{\cglsp1}{\@gls@hyp@opt\@cglsp1}
\newcommand*{\@cglsp1}[2][{}]{%
  \new@ifnextchar[{\@cglsp1@{#1}{#2}}{\@cglsp1@{#1}{#2}[]}]%
}
\def\@cglsp1@#1#2[#3]{%
  \GlossariesWarning{\string\cglsp1\space is defaulting to
    \string\glsp1\space since you haven't enabled entry counting}%
  \@glspl@{#1}{#2}[#3]%
}
\newcommand*{\cglsp1format}[2]{%
  \ifglshaslong{#1}{\glstentrylongpl{#1}}{\glstentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\glsmfuaddmap{\cglsp1}{\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*{\glstentryfmt}{\glsgenentryfmt}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglused\glslabel
        {%
          \glstentryplural{\glslabel}\glsinsert
        }%
        {%
          \glstentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
  }%
  \ifglused\glslabel

```

```

    {%
      \Glsentryplural{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirstplural{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glsuppercase{\glsentryplural{\glslabel}\glsinsert}%
    }%
    {%
      \glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%
    }%
  }%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glsentrytext{\glslabel}\glsinsert
    }%
    {%
      \glsentryfirst{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
      \Glsentryfirst{\glslabel}\glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glsuppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
    {%
      \glsuppercase{\glsentryfirst{\glslabel}\glsinsert}%
    }%
  }%
}%
{%
  \glscustomtext\glsinsert
}

```

```

}%
}
\newcommand*{\glsnacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\glsupercase
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
}%
}%
}%
\glscapscase
{%
\acronymfont{\glsentryshort{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshort{\glslabel}}\glsinsert
}%
{%
\glsupercase
{\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
}%
}%
}%
\glsifplural
{%
\glscapscase
{%
\genplacrformat{\glslabel}{\glsinsert}%
}%
{%
\Genplacrformat{\glslabel}{\glsinsert}%
}%
{%
\glsupercase
{\genplacrformat{\glslabel}{\glsinsert}}%
}%
}%
}

```

```

    {%
      \glscapscase
      {%
        \genacrfullformat{\glslabel}{\glsinsert}%
      }%
      {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
      }%
      {%
        \glsupercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
      }%
    }%
  }%
  {%
    \glscustomtext
  }%
}
\newcommand*{\genacrfullformat}[2]{%
  \glsentrylong{#1}#2\space
  (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\newcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}
\define@key{glslink}{format}{%

```

```

\def\@glsnumberformat{#1}
\newcommand*{\@gls@glslink@hyper@update@hook}{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef\hyperlink{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset@glslocalunset
  \else
    \let\@gls@do@glsunset@glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}{%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
  }
}
\newcommand*{\s@gls@hyp@opt}[1][{}]{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][{}]{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3][{}]{%
  \glsdoifexistsordo{#2}{%
    {%
      \let\do@gls@link@checkfirsthyper\relax
      \@gls@link[#1]{#2}{#3}%
    }{%
      \glstextformat{#3}%
    }%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@@link
}
\newcommand*{\@Gls@@link}[3][{}]{%
  \@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}

```

```

\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}{}}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glslinkprefix\glslabel}{\glstextformat{#3}}%
  \else
    \glsdonohyperlink{\glslinkprefix\glslabel}{\glstextformat{#3}}%
  \fi
}

```

```

\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
\@gls@restore@glslocal
}
\newcommand*\glsencapwrcontent}[1]{#{#1}}
\newcommand*\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*\@gls@saveentrycounter}{%
\def\@gls@Hcounter{}%
\ifthenelse{\equal{\@gls@counter}{equation}}%
{
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglsentrycounter}%
}%
{%
\def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
}%
\protected@edef\theHglentrycounter{\@gls@Hcounter}%
\restorecounters@
\fi
}%
}%
{%
\edef\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
\fi

```

```

}
\def\@set@glo@numformat#1#2#3#4{%
  \expandafter\@glo@check@mkidxrangear#3\@nil
  \protected@edef#1{%
    \glo@prefix setentrycounter[#4]{#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangear#1#2\@nil{%
\if#1\relax
  \def\@glo@prefix{()}%
  \if\relax#2\relax
    \def\@glo@suffix{glsnumberformat}%
  \else
    \def\@glo@suffix{#2}%
  \fi
\else
  \if#1\relax
    \def\@glo@prefix{}%
    \if\relax#2\relax
      \def\@glo@suffix{glsnumberformat}%
    \else
      \def\@glo@suffix{#2}%
    \fi
  \else
    \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
  \fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
  \def\@gls@checkedmkidx{}%
  \let\gls@xdystring=#1\relax
  \@onelevel@sanitize\gls@xdystring
  \edef\do@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\@null}%
  \do@gls@xdycheckbackslash
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \def\@gls@checkedmkidx{}%
  \expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\@null
  \expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
  \ifgls@esclocations
    \@for\@gls@tmp:=\gls@protected@pagefmts\do
    {%
      \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
      \@onelevel@sanitize\@gls@sanitized@tmp
      \edef\gls@dostubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
    }%
  \fi
}

```

```

        \gls@dostsubst
      }%
    \fi
    \let#1=\gls@xdystring
  }
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil""\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil\?\?\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \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}%
        \fi
    \fi
}

```

```

\def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2|\!\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

\@gls@quotechar\@gls@levelchar}%
\ifx\@null#3\@null
\def\@gls@checklevel{\@gls@checklevel#2!\@null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\@null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@gls@checkactual{\@gls@checkactual#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\@null#3\@null
\def\@gls@checkactual{\@gls@checkactual#2??\@null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\@null}%
\fi
\fi
\@gls@checkactual
}
\bgroup
\catcode'\@0\relax
\catcode'\@12\relax
\gdef\@gls@checkesc#1\#2\#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesc{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
\def\@gls@checkesc{\@gls@checkesc#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@backslashchar}%

```

```

        |ifx|null#3|null
          |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
        |else
          |def|@gls@checkesc{|@gls@checkesc#2|#3|null}%
        |fi
      |fi
    |@gls@checkesc
  }
|group
\def\@gls@xdycheckquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@xdycheckquote{\relax}%
    |else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \string"\string"}%
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
    |fi
  |else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \string"}%
    \ifx\null#3\null
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
    |else
      \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
    |fi
  |fi
  \@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={#1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand|else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand|fi

```

```

\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength@gls@tmplen
\newcommand*\glsdohypertarget}[2]{%
\glsdoshowtarget{#1}%
{%
\glsmeasureheight{gls@tmplen}{#2}%
\glsdohypertargethook{#1}{#2}%
\raisebox{gls@tmplen}{\hypertarget{#1}{}}%
#2%
}%
}
\newcommand*\glsdohypertargethook}[2]{}
\NewDocumentCommand\glslabelhypertarget{mm}{%
\protected@write\@auxout
{\glslabelhypertargetdefs}%
{%
\string\newlabel
{glslabelhypertargetprefix#1}%
{%
{\glslabelhypertargetvalue}{\thepage}{#2}{#1}{}}%
}%
}
}
\ExplSyntaxOn
\newcommand\glslabelhypertargetvalue{
\bool_lazy_and:nnF
{ \tl_if_exist_p:N \glscurrententrylabel }
{ \bool_not_p:n { \tl_if_empty_p:N \glscurrententrylabel } }
{
\glsentryname \glscurrententrylabel
}
}
}

```

```

\ExplSyntaxOff
\newcommand*\glslabelhypertargetprefix{}
\newcommand*\glslabelhypertargetdefs{%
  \let\glossentryname\glsentryname
  \let\glossentrydesc\glsentrydesc
  \let\glossentrysymbol\glsentrysymbol
  \let\Glossentryname\glsentryname
  \let\Glossentrydesc\glsentrydesc
  \let\Glossentrysymbol\glsentrysymbol
  \let\glossentrynameother\@gls@entry@field
}
\newcommand*\glsdohyperlink}[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsdohyperlinkhook{#1}{#2}%
    \hyperlink{#1}{#2}%
  }%
}
\newcommand*\glsdohyperlinkhook}[2]{}
\newcommand*\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand*\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand*\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*\gls}{\@gls@hyp@opt\@gls}
\newcommand*\@gls}[2] [] {%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2} [] }%
}

```

```

\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2} []}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2} []}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glsescapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
}

```

```

\glspostlinkhook
}
\newrobustcmd*{\glsp1}{\@gls@hyp@opt\@glsp1}
\newcommand*{\@glsp1}[2] []{%
  \new@ifnextchar[{\@glsp1@{#1}{#2}}{\@glsp1@{#1}{#2} []}%
}
\def\@glsp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLsp1}{\@gls@hyp@opt\@GLsp1}
\glsmfuaddmap{\glsp1}{\GLsp1}
\newcommand*{\@GLsp1}[2] []{%
  \new@ifnextchar[{\@GLsp1@{#1}{#2}}{\@GLsp1@{#1}{#2} []}%
}
\def\@GLsp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSp1}{\@gls@hyp@opt\@GLSp1}
\glsmfublocker{\GLSp1}
\newcommand*{\@GLSp1}[2] []{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2} []}%
}
\def\@GLSp1@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree

```

```

\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][{}%
\glsdoifexists{#2}{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\gls@scaps@case\@firstofthree
\def\glscustomtext{#3}%
\def\glsinsert{}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gls@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}%
\@glsdisp[#1]{#2}{\gls@sentence@case{#3}}%
}
\glsmfuaddmap{\glsdisp}{\Glsdisp}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\@gls@link[#1]{#2}{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
\newcommand*{\@glstext}[2][{}%
\new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]]}
\def\@glstext@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\gls@entrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\glsmfublocker{\GLStext}
\newcommand*{\@GLStext}[2][{}%
\new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]]}
\def\@GLStext@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\gls@supercase{\gls@entrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}

```

```

\glsmfuaddmap{\glstext}{\Glstext}
\newcommand*{@Glstext}[2] []{%
  \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2} []}]
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{@glsfirst}[2] []{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2} []}]
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfuaddmap{\glsfirst}{\Glsfirst}
\newcommand*{@Glsfirst}[2] []{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfublocker{\Glsfirst}
\newcommand*{@Glsfirst}[2] []{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2} []}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{@glsplural}[2] []{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{@Glsplural}[2] []{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfublocker{\Glsplural}
\newcommand*{@Glsplural}[2] []{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{@glsfirstplural}[2] []{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3]{%

```

```

    \@gls@field@link{#1}{#2}{\glentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][\%
    \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}]
\def\@Glsfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\glsmfublocker{\GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2][\%
    \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2}[]}]
\def\@GLSfirstplural@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\gluppercase{\glentryfirstplural{#2}#3}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2][\%
    \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2}[]}]
\def\@glsname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\glsmfuaddmap{\glsname}{\Glsname}
\newcommand*{\@Glsname}[2][\%
    \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2}[]}]
\def\@Glsname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\glsmfublocker{\GLSname}
\newcommand*{\@GLSname}[2][\%
    \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2}[]}]
\def\@GLSname@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\gluppercase{\glentryname{#2}#3}%
}
\newrobustcmd*{\glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2][\%
    \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2}[]}]
\def\@glsdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glentrydesc{#2}#3}%
}
\newrobustcmd*{\Glsdesc}{\@gls@hyp@opt\@Glsdesc}
\glsmfuaddmap{\glsdesc}{\Glsdesc}
\newcommand*{\@Glsdesc}[2][\%
    \new@ifnextchar[{\@Glsdesc@{#1}{#2}}{\@Glsdesc@{#1}{#2}[]}]
\def\@Glsdesc@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glentrydesc{#2}#3}%
}
\newrobustcmd*{\GLSdesc}{\@gls@hyp@opt\@GLSdesc}

```

```

\glsmfublocker{\GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar [ {\@GLSdesc@{#1}{#2}} {\@GLSdesc@{#1}{#2} [] } }
\def \@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2} {\glsuppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\glsdescplural} {\@gls@hyp@opt \@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar [ {\@glsdescplural@{#1}{#2}} {\@glsdescplural@{#1}{#2} [] } }
\def \@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2} {\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\Glsdescplural} {\@gls@hyp@opt \@Glsdescplural}
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
\newcommand*{\@Glsdescplural}[2] [] {%
  \new@ifnextchar [ {\@Glsdescplural@{#1}{#2}} {\@Glsdescplural@{#1}{#2} [] } }
\def \@Glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2} {\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural} {\@gls@hyp@opt \@GLSdescplural}
\glsmfublocker{\GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar [ {\@GLSdescplural@{#1}{#2}} {\@GLSdescplural@{#1}{#2} [] } }
\def \@GLSdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2} {\glsuppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol} {\@gls@hyp@opt \@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar [ {\@glssymbol@{#1}{#2}} {\@glssymbol@{#1}{#2} [] } }
\def \@glssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2} {\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol} {\@gls@hyp@opt \@Glsymbol}
\glsmfuaddmap{\glssymbol}{\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}
\glsmfublocker{\GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
  \new@ifnextchar [ {\@GLSsymbol@{#1}{#2}} {\@GLSsymbol@{#1}{#2} [] } }
\def \@GLSsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2} {\glsuppercase{\glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural} {\@gls@hyp@opt \@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
  \new@ifnextchar [ {\@glssymbolplural@{#1}{#2}} {\@glssymbolplural@{#1}{#2} [] } }
\def \@glssymbolplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\glsmfuaddmap{\glsymbolplural}{\Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}]
}
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\glsmfublocker{\GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2] [] {%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2} []}]
}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}]
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\glsmfuaddmap{\glsuseri}{\GLSuseri}
\newcommand*{\@GLSuseri}[2] [] {%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2} []}]
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}

```

```

\glsmfublocker{\GLSuserii}
\newcommand*{\@GLSuserii}[2] [] {%
  \new@ifnextchar [{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2} []}]
\def\@GLSuserii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2] [] {%
  \new@ifnextchar [{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2} []}]
\def\@glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\glsmfuaddmap{\glsuseriii}{\Glsuseriii}
\newcommand*{\@Glsuseriii}[2] [] {%
  \new@ifnextchar [{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2} []}]
\def\@Glsuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\glsmfublocker{\GLSuseriii}
\newcommand*{\@GLSuseriii}[2] [] {%
  \new@ifnextchar [{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2} []}]
\def\@GLSuseriii@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2] [] {%
  \new@ifnextchar [{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} []}]
\def\@glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\glsmfuaddmap{\glsuseriv}{\Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar [{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} []}]
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\glsmfublocker{\GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} []}]
\def\@GLSuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}]
\def\@glsuserv@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\glsmfuaddmap{\glsuserv}{\Glsuserv}
\newcommand*{\@Glsuserv}[2][\%
\new@ifnextchar[{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2}[]}]
\def\@Glsuserv@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\glsmfublocker{\GLSuserv}
\newcommand*{\@GLSuserv}[2][\%
\new@ifnextchar[{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2}[]}]
\def\@GLSuserv@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsuppercase{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2][\%
    \new@ifnextchar[{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2}[]}]
\def\@glsuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\glsmfuaddmap{\glsuservi}{\GLSuservi}
\newcommand*{\@GLSuservi}[2][\%
    \new@ifnextchar[{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2}[]}]
\def\@GLSuservi@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\ns@acrshort}
\newcommand*{\ns@acrshort}[2][\%
    \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2}[]}]
}
\def\@acrshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \let\glsifplural\@secondoftwo
        \let\glsescapscase\@firstofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsentryshort{#2}}#3%
        }%
    }%
}

```

```

    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\ns@Acrshort}
\glsmfuaddmap{\Acrshort}{\Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsifscaps\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
\def\@ACRshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsifscaps\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\Glsentryshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
}
\def\@acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%

```

```

{%
  \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \def\glslabel{#2}%
  \let\gl@ifplural\@firstoftwo
  \let\glscapscase\@firstofthree
  \let\gl@insert\@empty
  \def\glscustomtext{%
    \acronymfont{\gl@entryshortpl{#2}}#3%
  }%
  \@gl@link[#1]{#2}{\csname gl@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gl@hyp@opt\ns@Acrshortpl}
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2][]{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\gl@insert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gl@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gl@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][]{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\gl@ifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\gl@insert\@empty
    \def\glscustomtext{%
      \gl@supercase{\acronymfont{\gl@entryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gl@\glstype @entryfmt\endcsname}%
  }%
}

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\ns@acrlong}
\newcommand*{\ns@acrlong}[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@firstofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlong}{\@gls@hyp@opt\ns@Acrlong}
\glsmfuadddmap{\Acrlong}{\Acrlong}
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\gls caps case\@secondofthree
    \let\glsinsert\@empty
    \def\gls custom text{%
      \Glsentrylong{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\glsmfublocker{\ACRlong}
\newcommand*{\ns@ACRlong}[2][{}]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \glsuppercase{\glsentrylong{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\AcrLongpl}{\@gls@hyp@opt\ns@AcrLongpl}
\glsmfuaddmap{\AcrLongpl}{\AcrLongpl}
\newcommand*{\ns@AcrLongpl}[2][{}]{%
  \new@ifnextchar[{\@AcrLongpl{#1}{#2}}{\@AcrLongpl{#1}{#2}[]]}%
}
\def\@AcrLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}

```

```

}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2} []}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glsifscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\gls@type @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\glsentencecase\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ???
  }%
}
\newcommand{\@Glsentryfield}[2]{%
  \gls@text@pdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\ifdef\text@pdfstring

```

```

{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
\newcommand*{\glstentryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{name}}}%
}
\glsmfuexcl{\Glsentryname}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newrobustcmd*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \protected@edef\@glo@text{\@Gls@entry@field{#1}{name}}%
    \expandafter\glssentencecase\expandafter{\@glo@text}%
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\@org@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{\@nil
    \expandafter\ifx\@gls@body\glstentrylong\relax
      \expandafter\Glsentrylong\@gls@rest
    \else
      \expandafter\ifx\@gls@body\glstentryshort\relax
        \expandafter\Glsentryshort\@gls@rest
      \else
        \expandafter\ifx\@gls@body\acronymfont\relax
          {%
            \let\glstentryshort\Glsentryshort
            \@glo@text
          }%
        \else
          \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
            {%
              \let\glstentryshort\Glsentryshort
              \@glo@text
            }%
          \else
            \expandafter\glssentencecase\expandafter{\@glo@text}%
          \fi
        \fi
      \fi
    \fi
  }%
}

```

```

    {%
      \@Gls@entry@field{#1}{name}%
    }%
  }
  \newcommand*\glsentrydesc[1]{\@gls@entry@field{#1}{desc}}
  \newcommand*\Glsentrydesc[1]{\@Glsentryfield{#1}{desc}}
  \glsmfuexcl{\Glsentrydesc}
  \newcommand*\glsentrydescplural[1]{%
    \@gls@entry@field{#1}{descplural}%
  }
  \newcommand*\Glsentrydescplural[1]{%
    \@Glsentryfield{#1}{descplural}%
  }
  \glsmfuexcl{\Glsentrydescplural}
  \newcommand*\glsentrytext[1]{\@gls@entry@field{#1}{text}}
  \newcommand*\Glsentrytext[1]{%
    \@Glsentryfield{#1}{text}%
  }
  \glsmfuexcl{\Glsentrytext}
  \newcommand*\glsentryplural[1]{%
    \@gls@entry@field{#1}{plural}%
  }
  \newcommand*\Glsentryplural[1]{%
    \@Glsentryfield{#1}{plural}%
  }
  \glsmfuexcl{\Glsentryplural}
  \newcommand*\glsentrysymbol[1]{%
    \@gls@entry@field{#1}{symbol}%
  }
  \newcommand*\Glsentrysymbol[1]{%
    \@Glsentryfield{#1}{symbol}%
  }
  \glsmfuexcl{\Glsentrysymbol}
  \newcommand*\glsentrysymbolplural[1]{%
    \@gls@entry@field{#1}{symbolplural}%
  }
  \newcommand*\Glsentrysymbolplural[1]{%
    \@Glsentryfield{#1}{symbolplural}%
  }
  \glsmfuexcl{\Glsentrysymbolplural}
  \newcommand*\glsentryfirst[1]{%
    \@gls@entry@field{#1}{first}%
  }
  \newcommand*\Glsentryfirst[1]{%
    \@Glsentryfield{#1}{first}%
  }
  \glsmfuexcl{\Glsentryfirst}
  \newcommand*\glsentryfirstplural[1]{%
    \@gls@entry@field{#1}{firstpl}%
  }
}

```

```

\newcommand*\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
\glsmfuexcl{\Glsentryfirstplural}
\newcommand*\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}
\newcommand*\glstentrytitlecase}[2]{%
  \glstexorpdfstring
  {\@Glsentrytitlecase{#1}{#2}}%
  {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\newrobustcmd*\@Glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\newcommand*\glstentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*\glstentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*\glstentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*\glstentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuexcl{\Glsentryuseri}
\newcommand*\glstentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newcommand*\Glsentryuserii}[1]{%
  \@Glsentryfield{#1}{userii}%
}
\glsmfuexcl{\Glsentryuserii}
\newcommand*\glstentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newcommand*\Glsentryuseriii}[1]{%
  \@Glsentryfield{#1}{useriii}%
}
\glsmfuexcl{\Glsentryuseriii}
\newcommand*\glstentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
}

```

```

\newcommand*\Glsentryuseriv}[1]{%
  \@Glsentryfield{#1}{useriv}%
}
\glsmfuexcl{\Glsentryuseriv}
\newcommand*\glsentryuser}[1]{%
  \@gls@entry@field{#1}{user}%
}
\newcommand*\Glsentryuserv}[1]{%
  \@Glsentryfield{#1}{userv}%
}
\glsmfuexcl{\Glsentryuserv}
\newcommand*\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newcommand*\Glsentryuserivi}[1]{%
  \@Glsentryfield{#1}{userivi}%
}
\glsmfuexcl{\Glsentryuserivi}
\newcommand*\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newcommand*\Glsentryshort}[1]{%
  \@Glsentryfield{#1}{short}%
}
\glsmfuexcl{\Glsentryshort}
\newcommand*\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newcommand*\Glsentryshortpl}[1]{%
  \@Glsentryfield{#1}{shortpl}%
}
\glsmfuexcl{\Glsentryshortpl}
\newcommand*\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuexcl{\Glsentrylong}
\newcommand*\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuexcl{\Glsentrylongpl}
\newcommand*\glsentryfull}[1]{%
  \glsentrylong{#1} (\glsentryshort{#1})%
}
\newcommand*\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\Glsentryshort{#1})%
}
\glsmfuexcl{\Glsentryfull}
\newcommand*\glsentryfullpl}[1]{%
  \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glsentryshortpl{#1})%
}

```

```

}
\glsmfueycl{\Glsentryfullpl}
\newcommand*{\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
}
}
\newcommand*{\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \bgroup
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \let\@org@glsnumberformat\glsnumberformat
    \def\glsnumberformat##1{##1}%
    \protected@edef\the@numberlist{%
      \csname glo@\@glo@label @numberlist\endcsname}%
    \def\@gls@numlist@sep{}%
    \def\@gls@numlist@nextsep{}%
    \def\@gls@numlist@lastsep{}%
    \def\@gls@thislist{}%
    \def\@gls@donext@def{}%
    \renewcommand\do[1]{%
      \protected@edef\@gls@thislist{%
        \@gls@thislist
        \noexpand\@gls@numlist@sep
        ##1%
      }%
      \let\@gls@numlist@sep\@gls@numlist@nextsep
      \def\@gls@numlist@nextsep{\glsnumlistsep}%
      \@gls@donext@def
      \def\@gls@donext@def{%
        \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
      }%
    }%
    \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
    \let\@gls@numlist@sep\@gls@numlist@lastsep
    \@gls@thislist
  }%
}

```

```

        \egroup
      }%
    }
  }
  \newcommand*\glsnumlistsep}{, }
  \newcommand*\glsnumlistlastsep}{ \& }
  \newcommand*\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
    \def\@glo@label{#2}%
    \@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
  \define@key{glossadd}{counter}{\def\@gls@counter{#1}}
  \define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
  \define@key{glossadd}{types}{\def\@glo@type{#1}}
  \newrobustcmd*\glsadd}[2][]{%
    \@glsadjustmode
    \glsdoifexists{#2}%
    {%
      \def\@glsnumberformat{glsnumberformat}%
      \protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
      \setkeys{glossadd}{#1}%
      \@gls@saveentrycounter
      \@gls@setsort{#2}%
      \@do@wrglossary{#2}%
    }%
  }
  \glsmfuexcl{\glsadd}
  \newcommand*\@glsadjustmode}{
  \AtBeginDocument{\renewcommand*\@glsadjustmode}{\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\gls closebrace{\expandafter\@gobble\string\}}
  \edef\glsbackslash{\expandafter\@gobble\string\ \}
  \edef\glsquote#1{\string"#1\string"}
  \edef\glspercentchar{\expandafter\@gobble\string\%}
  \edef\gls tildechar{\string~}
  \ifglsxindy
    \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]{%
    \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglxindy
  \newcommand*\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*\GlsSetXdyNumberGroupOrder}[1]{%
  \ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglxindy
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
  }
  \newcommand*\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnoxywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*\@glsminrange}{2}
\ifglxindy
  \newcommand*\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*\@glsminrange}{#1}}
\else
  \newcommand*\GlsSetXdyMinRangeLength}[1]{%
    \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglxindy
\def\writeist{%
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \gls@addpredefinedattributes
  \openout\glswrite=\istfilename

```

```

\write\glswrite{;; xindy style file created by the glossaries
  package}%
\write\glswrite{;; for document '\jobname' on
  \the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
  \ifx\@xdystyle\@empty
  \else
  \protected@write\glswrite{}{(require
    \string"\@xdystyle.xdy\string")}%
  \fi
}%
\write\glswrite{^^J%
  ; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@gls@classI:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}{(define-location-class
    \string"\@gls@classI\string"^^J\space\space\space
    (
      :sep "{{"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
  \protected@write\glswrite{}{(define-location-class
    \string"\@gls@classII-\@gls@classI\string"
    ^^J\space\space\space
    (
      :sep "{"
      \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
      :sep "}"
      \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
      :sep "}"
    )
    ^^J\space\space\space
    :min-range-length \@glsminrange^^J%
  )
}%
}%
\write\glswrite{^^J; user defined location classes}%

```

```

\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
: class \string"see\string"^^J\space\space\space
: open \string"\string\glsseeformat\string"
: close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
: open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{ }\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glspercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)%}
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glsresetentrylist
\glstildechar n\string")}%
\write\glswrite{(markup-locclass-list :open
\string"\glsopenbrace\string\glossaryentrynumbers
\glsopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"

```

```

        :close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string"\string\glsgroupheading
  \glsopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string"}}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\else
  \edef\@gls@actualchar{\string?}
  \edef\@gls@encapchar{\string|}
  \edef\@gls@levelchar{\string!}
  \edef\@gls@quotechar{\string"}%
  \let\GlsSetQuote\gls@nosetquote
  \def\writeist{\relax
    \ifundef{\glswrite}{\newwrite\glswrite}{}\relax
    \openout\glswrite=\istfilename
    \write\glswrite{\glspersentchar\space makeindex style file
      created by the glossaries package}
    \write\glswrite{\glspersentchar\space for document
      '\jobname' on \the\year-\the\month-\the\day}
    \write\glswrite{actual '\@gls@actualchar'}
  }

```

```

\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string"}
\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else

```

```

        \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
    \fi
    \@gls@writeisthook
    \closeout\glswrite
    \let\writeist\relax
    \let\noist\@gls@noist@toolate
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{}
\ifglsxindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\@gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\@ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1####2####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else

```

```

\noexpand\def\noexpand\@gls@checkquote{%
  \noexpand\@gls@checkquote###2#1###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
  \noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkquote###1\noexpand\@nil
  #1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesquote###1\noexpand\@nil
  \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesactual###1\noexpand\@nil
  \noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkactual###1\noexpand\@nil
  \noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
  \noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkeschar###1\noexpand\@nil
  \noexpand\|\noexpand\|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
  \noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
  \noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescllevel###1\noexpand\@nil
  \noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked

```

```

        \noexpand\@gls@checkedmkidx{###1}%
    \noexpand\def\noexpand\@gls@checkedmkidx{%
    \noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
    \@backslashchar\@backslashchar\noexpand\null
    \noexpand\expandafter\noexpand\@gls@updatechecked
    \noexpand\@gls@checkedmkidx{###1}%
    }%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}

```

```

\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
  {\string\GlsSetQuote\space not permitted here}%
  {Move \string\GlsSetQuote\space earlier in the preamble, as
  soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*{\@gls@extramakeindexopts}[1]{}
\newcommand{\noist}{%
  \@gls@addpredefinedattributes
  \let\writeist\relax
}
\newcommand{\@gls@noist@toolate}{%
  \GlossariesWarning{too late to use \string\noist. The style file
  has already been written}%
}
\newcommand*{\@makeglossary}[1]{%
  \ifglossaryexists{#1}%
  {%
    \ifglssavewrites
      \expandafter\newtoks\csname glo@#1@filetok\endcsname
    \else
      \expandafter\newwrite\csname glo@#1@file\endcsname
      \expandafter\@glsopenfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%
    \PackageError{glossaries}%
    {Glossary type ‘#1’ not defined}%
    {New glossaries must be defined before using \string\makeglossaries}%
  }%
}
\newcommand*{\@glsopenfile}[2]{%
  \immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
  \PackageInfo{glossaries}{Writing glossary file
  \jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
  \closeout\csname glo@#1@file\endcsname
}
\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
\ifglsxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\gls@xdy@set@lang{#1}%
\ifdefvoid\gls@codepage

```

```

        {\let\@gls@codepage\@empty}%
        {\def\@gls@codepage{-C \gls@codepage\space}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
\@gls@codepage
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@types}{\@glo@types}
\newcommand{\@gls@automake@immediate}{%

```

```

\ifnum\gls@automake@nr=2\relax
  \protected@edef\gls@tmp{\@gls@automake@types}%
  \@for\@gls@type:=\gls@tmp\do{%
    \ifdefempty{\@gls@type}{}%
      {\@gls@automake@immediate{\@gls@type}}%
    }%
  \glsautomakefalse
  \renewcommand*{\@gls@doautomake}{}%
\else
  \ifdefempty\gls@automake@makegloss{}%
    {%
      \IfFileExists{\jobname.aux}%
        {\immediate\write18{\@gls@automake@makegloss\space \jobname}}{}%
      \glsautomakefalse
      \renewcommand*{\@gls@doautomake}{}%
    }%
  \fi
}
\newcommand*{\makeglossaries}{%
  \ifglxindy
    \def\glsindexingsetting{xindy}%
  \else
    \def\glsindexingsetting{makeindex}%
  \fi
  \@domakeglossaries
  {%
    \@gls@@automake@immediate
    \ifundef{\glswrite}{\newwrite\glswrite}{}%
    \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
    \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
    \ifundef\@@gls@extramakeindexopts
    {}%
    {%
      \protected@write\@auxout{}{\string\providecommand
        \string\@gls@extramakeindexopts[1]{} }
      \protected@write\@auxout{}{\string\@gls@extramakeindexopts
        {\@@gls@extramakeindexopts}}%
    }%
    \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
    \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
    \@for\@glo@type:=\@glo@types\do{%
      \ifthenelse{\equal{\@glo@type}{}}{}{%
        \@makeglossary{\@glo@type}}%
      }%
    \renewcommand*\newglossary[4] []{%
      \PackageError{glossaries}{New glossaries
        must be created before \string\makeglossaries}{You need
        to move \string\makeglossaries\space after all your
        \string\newglossary\space commands}}%
    \let\@makeglossary\@gobble

```

```

\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\ifglssavenumberlist
  \edef\@gls@dodolistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \protected@edef\gls@tmp{\@gls@automake@types}%
    \@for\@gls@type:=\gls@tmp\do{%
      \ifdefempty{\@gls@type}{%
        {\@gls@automake{\@gls@type}}%
      }%
    }%
  }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,%}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@nofwarn}[1]{%
      \GlossariesWarning{Empty glossary for

```

```

\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
\let\@gls@checkmkidxchars\@gobble
\glsesclocationsfalse
\let\@do@wrglossary\gls@noidxglossary
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\@gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
  \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@noprntglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
or \string\printnoidxglossaries ^^J
found. (Remove \string\makenoidxglossaries\space if you
don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
defined in the preamble^^Jwhen you use
\string\makenoidxglossaries}%
{Either move your definitions to the preamble or use
\string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%

```

```

??\glsdoifexists{##1}%
{%
  \GlossariesWarning{Missing location list for ‘##1’. Either
    a rerun is required or you haven’t referenced the entry}%
}%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}

```

```

\let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
\let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@wrglossdisablelocationcmds}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}%
      }%
    }%
}
}
\AtEndDocument{%
  \gls@writefiles
}
\newcommand*\@gls@writefiles}{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}}%
    }%
}

```

```

    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\glo@type}%
      \immediate\write\glswrite{%
        \expandafter\the
          \csname glo@\glo@type @filetok\endcsname}%
      \immediate\closeout\glswrite
    \fi
  }%
}
\newcommand*{\gls@glossary}[1]{%
  \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp^^J}%
  \else
    \ifcsdef{glo@#1@file}%
      {%
        \expandafter\protected@write\csname glo@#1@file\endcsname{%
          \gls@disablepagerefexpansion}{#2}%
        }%
      {%
        \ifignoredglossary{#1}{}%
          {%
            \GlossariesWarning{No file defined for glossary '#1'}%
          }%
        }%
      }%
    \fi
  \fi
}

```

```

    }%
  }%
\fi
\endgroup\@esphack
}
\newcommand*\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*\glswriteentry[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*\gls@disablepagerefexpansion{%
\glswrglossdisablelocationcmds
\ifglsesclocations
  \@for\@gls@this:=\gls@protected@pagefmts\do
  {%
    \expandafter\let\@gls@this\relax
  }%
\fi
}
\newcommand{\glswrglossdisablelocationcmds}{}
\newcommand*\gls@alphpage}{\@alph\c@page}
\newcommand*\gls@Alphpage}{\@Alph\c@page}
\newcommand*\gls@numberpage}{\number\c@page}
\newcommand*\gls@arabicpage}{\@arabic\c@page}
\newcommand*\gls@romanpage}{\romannumeral\c@page}
\newcommand*\gls@Romanpage}{\@Roman\c@page}
\newcommand*\glsaddprotectedpagefmt}[1]{%
  \ifglsesclocations
    \protected@eappto\gls@protected@pagefmts{\, \expandonce{\csname gls#1page\endcsname}}%
    \csedef{gls#1page}{\, \expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gls#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}

```

```

\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@do@wrglossary}[1]{%
  \ifglseclocations
    \@do@esc@wrglossary{#1}%
  \else
    \@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand{\glswrglosslocationtarget}[1]{#1}
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
\newcommand*{\@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
    \endgroup

```

```

\def\number{\expandafter\gls@number}%
\def\romannumeral##1{%
  \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
\fi
\def\@arabic##1{%
  \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\@Roman##1{%
  \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
  \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
  \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theglsentrycounter\theglsentrycounter\relax
\def\@glo@counterprefix{}%
\else
\protected@edef\@glsHlocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@glsHlocref
\@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
\protected@edef\@gls@label{\glsdetoklabel{#1}}%
\@do@wrglossary
}
\newcommand*{\@do@wrglossary}{%
\ifglxindy
\expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
\def\@glo@range{}%
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:open-range}%
\else
\expandafter\if\@glo@prefix\relax
\def\@glo@range{:close-range}%
\fi
\fi
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
:locref \string{"\@glo@counterprefix}{\@glslocref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname

```

```

        \@gls@encapchar\@glo@numfmt}{\@gls@locref}}%
    \fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
  \bgroup
  \gls@wrgloss@disable@anchor@cmds
  \protected@edef\@do@gls@getcounterprefix{%
    \noexpand\egroup
    \noexpand\@gls@getcounterprefix{#1}{#2}%
  }%
  \@do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
  \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
    \ifx\@glo@counterprefix\@empty
      \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
        prefixing location ‘#1’. You need to modify the
        definition of \string\theH\@gls@counter otherwise you
        will get the warning: “name{\@gls@counter.#1}’ has been
        referenced but does not exist”}%
    \fi
  \fi
}
\newcommand{\@do@seeglossary}[2]{%
  \def\@gls@xref{#2}%
  \@onelevel@sanitize\@gls@xref
  \@gls@checkmkidxchars\@gls@xref
  \ifglsxindy
    \gls@glossary{\csname glo@#1@type\endcsname}{%
      (indexentry
        :key (\csname glo@#1@index\endcsname)
        :xref (\string"\@gls@xref\string")
        :attr \string"see\string"
      )
    }%
  \else
    \gls@glossary{\csname glo@#1@type\endcsname}{%
      \string\glossaryentry{\csname glo@#1@index\endcsname

```

```

\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
\do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
\emph{#1} \glsseeelist{#2}}
\newrobustcmd*\glsseeelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext
\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseeelastsep
\let\@gls@donext\glsseesep
}%
}
\newcommand*\glsseeelastsep}{\space\andname\space}
\newcommand*\glsseeesep}{, }
\newrobustcmd*\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat}[1]{%
\ifglshasshort{#1}{\glsentrytext{#1}}{\glsentryname{#1}}%
}
\newcommand*\gls@save@numberlist}[1]{%
\ifglssavenumberlist
\toks@{#1}%
\protected@edef\@do@writeaux@info{%
\noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\@do@writeaux@info
\protected@write\@auxout{}{\@do@writeaux@info}%
\fi
}
\newcommand*\warn@noprntglossary}{}%

```

```

\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{}
\newcommand*{\@gls@preglossaryhook}{}
\newcommand{\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glo@type@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary ‘\@glo@type’. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi

```

```

}%
\def\gls@dotoctitle{\glssettoctitle{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\gls@printglossary@begin
\@printgloss@setsort
\setkeys{printgloss}{#1}%
\@printgloss@checkexists{\@glo@type}%
{%
\ifx\glossarytitle\org@glossarytitle
\else
\expandafter\let\csname @glo@type@\@glo@type @title\endcsname
\glossarytitle
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotoctitle
\@glossarystyle
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
\gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\gls@printglossary@end
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\gls@printglossary@begin}{\begingroup}
\newcommand{\gls@printglossary@end}{%
\expandafter\endgroup\if@endpe\@doendpe\fi
}
\newcommand{\@print@glossary}{%
\makeatletter
\@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
\IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
{}%
{\null}%
\ifglsxindy
\gls@xdy@set@lang{\@glo@type}%
\edef\@do@auxoutstuff{%

```

```

        \noexpand\AtEndDocument{%
            \noexpand\immediate\noexpand\write\@auxout{%
                \string\providecommand\string\@xdylanguage[2]{}%
            \noexpand\immediate\noexpand\write\@auxout{%
                \string\@xdylanguage{\@glo@type}{\@gls@langmod}}%
        }%
    }%
    \do@auxoutstuff
    \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
            \noexpand\immediate\noexpand\write\@auxout{%
                \string\providecommand\string\@gls@codepage[2]{}%
            \noexpand\immediate\noexpand\write\@auxout{%
                \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
        }%
    }%
    \do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
    \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
    \glosortentrieswarning
    \def\@glo@sortinglist{}%
    \def\@glo@sortinghandler{#1}%
    \protected@edef\@glo@type{#2}%
    \forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
    \csdef{\@glsref@#2}{}%
    \@for\@this@label:=\@glo@sortinglist\do{%
        \xifinlistcs{\@this@label}{\@glsref@#2}%
        {}%
        {%
            \listcsxadd{\@glsref@#2}{\@this@label}%
        }%
        \ifcsdef{\@glo@sortingchildren@\@this@label}%
        {%
            \@glo@addchildren{#2}{\@this@label}%
        }%
        {}%
    }%
}
\newcommand*{\@glo@addchildren}[2]{%
    \bgroup
    \letcs{\@glo@childlist}{\@glo@sortingchildren@#2}%
    \@for\@this@childlabel:=\@glo@childlist\do
    {%
        \xifinlistcs{\@this@childlabel}{\@glsref@#1}%
        {}%
    }%
}

```

```

        {%
        \listcsxadd{@glsref{#1}}{\@this@childlabel}%
        }%
        \ifcsdef{@glo@sortingchildren@ \@this@childlabel}%
        {%
        \glo@addchildren{#1}{\@this@childlabel}%
        }%
        {%
        }%
        }%
    \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
    \ifglshasparent{#1}%
    {%
    \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
    \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    }%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {%
    }%
    {%
    \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
    }%
    {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
    }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
    \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
    \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef@glo@do@compare{%
        \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce@gls@sort@B}%
        {\expandonce@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
    \letcs@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%

```

```

\edef\glo@do@compare{%
  \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
  {\expandonce\@gls@sort@B}%
  {\expandonce\@gls@sort@A}%
}%
\glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{ }%
  }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{ }%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \ifcsdef{@glo@sorthandler@glorder}%
  {%
    \@glo@sortentries{\cuse{@glo@sorthandler@glorder}}{#1}%
  }%
  {%
    \PackageError{glossaries}{Unknown sort handler ‘glorder’}{}%
  }%
}%
}
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
  \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
  \string\printnoidxglossary[sort=standard]}{}%
}%
}
}
\newcommand*\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=case]}{}%
  }%
}
}
\newcommand*\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
  }%
}
}
\newcommand*\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{@glsref@#1}%
    {%
      \listead{\@glo@sortinglist}{\@gls@thislabel}%
    }%
  }%
  {%
  }%
}
\cslet{@glsref@#1}{\@glo@sortinglist}%

```

```

}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\@glossarytoctitle]{\@glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
    \end{theglossary}%
    \glossarypostamble
  }%
  {}%
  \@gls@noref@warn{\@glo@type}%
}%
}
\def\@glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{}{}\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%

```

```

\ifdefempty\@glo@thislettergrp
{%
  \def\@glo@thislettergrp{glssymbols}%
}%
{%
  \count@=\uccode‘#1\relax
  \ifnum\count@=0\relax
    \def\@glo@thislettergrp{glssymbols}%
  \else
    \ifdefstring\@glo@sorttype{case}%
    {%
      \count@=#1\relax
    }%
    {%
    }%
    \edef\@glo@thislettergrp{\the\count@}%
  \fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
  \global\letcs{\@gls@loclist}{glo\@glsdetoklabel{#1}@loclist}%
  \ifglshasparent{#1}%
  {%
    \gls@level=\csuse{glo\@glsdetoklabel{#1}@level}\relax
    \ifdefvoid{\@gls@loclist}
    {%
      \subglossentry{\gls@level}{#1}{}%
    }%
    {%
      \subglossentry{\gls@level}{#1}%
    }%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}
{%
  \letcs{\@gls@sort}{glo\@glsdetoklabel{#1}@sort}%
  \expandafter\glo@grabfirst\@gls@sort{}{}\@nil
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \global\let\@glo@thislettergrp\@glo@thislettergrp
      \glsgroupskip
    }%
    \glsgroupheading{\@glo@thislettergrp}%
  }%
}
\global\let\@gls@currentlettergroup\@glo@thislettergrp

```

```

\ifdefvoid{\@gls@loclist}
{
  \glossentry{#1}{}%
}%
{
  \glossentry{#1}%
  {
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}
}
\newcommand*\glsnoidxloclist[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*\glsnoidxloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
}
\newcommand*\glsnoidxdisplayloclisthandler[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[1]{#2}%
  \csuse{#3}{#4}%
}
}
\newcommand*\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
  }%
  \ifcsundef{\@gls@ref@#1}{\csgdef{\@gls@ref@#1}{}}{%
  }%
  \ifinlistcs{#2}{\@gls@ref@#1}%
  {}%
  {\listcsgadd{\@gls@ref@#1}{#2}}%
}

```

```

\ifcsundef{glo@glstetoklabel{#2}@loclist}%
{\csgdef{glo@glstetoklabel{#2}@loclist}{}}%
{}%
\listcsgadd{glo@glstetoklabel{#2}@loclist}{#3}%
}%
}
\define@key{printgloss}{type}{\def@glo@type{#1}}
\define@key{printgloss}{title}{%
\def@glossarytitle{#1}%
\let@glstetoktitle\relax
}
\define@key{printgloss}{toctitle}{%
\def@glossarytoctitle{#1}%
\let@glstetoktitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{@glstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def@glossarystyle{\setglossarystyle{#1}}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\glstetoksection@val\glstetoksection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\glstetoksection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{\label{\glstetokprefix@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glstetokprefix@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glstetokgroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glstetokpostdot#1}%
}

```

```

}
\newcommand*\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glentrycounter#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}{%
  \ifglentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*\glsstepentry}[1]{%

```

```

\ifglstepcounter
  \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]{%
  \ifglstepcounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglstepcounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}
\newcommand*{\glsentryitem}[1]{%
  \ifglstepcounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glsstepsubentry{#1}\glssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%

```

```

\@gls@warnontheGLOSSdefined
\renewenvironment{theglossary}{}{}%
}
\newcommand*\glossaryheader{}
\newcommand*\glstarget}[2]{\@glstarget{\glo@linkprefix#1}{#2}}
\newcommand\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*\glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\letcs{\glo@name}{\glo@GLSDetoklabel{#1}@name}%
\expandafter\glsnamefont\expandafter{\glo@name}%
}%
}
\newcommand*\Glossentryname}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsnamefont{\Glsentryname{#1}}%
}%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*\glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsentrydesc{#1}%
}%
}
\newcommand*\Glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\Glsentrydesc{#1}%
}%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*\glossentrysymbol}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsentrysymbol{#1}%
}%
}
\newcommand*\Glossentrysymbol}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\Glsentrysymbol{#1}%
}%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*\glsgroupskip{}
\newcommand*\glsgroupheading}[1]{}
\newrobustcmd*\glsgetgrouptitle}[1]{%

```

```

    \@gls@getgrouptitle{#1}{\@gls@grptitle}%
    \@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
  \def#2{#1}%
  \regex_match:nnT { ^\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{}\letcs#2{#1groupname} }
}
\ExplSyntaxOff
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
  }%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][ ]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand{\newglossarystyle}[2]{%

```

```

\ifcsundef{@glsstyle@#1}%
{%
  \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
}%
{%
  \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{-}%
}%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{-}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glsnumber#1{#1}%
}%
{%
  \def\glsnumber#1{\xp@glsnumber{#1}}
}
\newrobustcmd{\xp@glsnumber}[1]{%
  \@glsnumber#1\nohyperpage{}\@nil
}
\def\@glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
  \else
    \gls@delimR#1\delimR\delimR\%
  \fi
  \ifx\#2\%
  \else
    #2%
  \fi
  \ifx\#3\%
  \else
    \@glsnumber#3\@nil
  \fi
}
\def\gls@delimR#1\delimR #2\delimR #3\%{
  \ifx\#2\%
    \gls@delimN{#1}%
  \else
    \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
  \fi
}
\def\gls@delimN#1{\gls@@delimN#1\delimN \delimN\}

```

```

\def\gls@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
\gls@numberlink{#1}%
\else
\gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glsentrycounter\@glo@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glsnumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glsnumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glsnumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glsnumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glsnumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glsnumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glsnumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glsnumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glsnumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
\def\gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
}

```

```

\ifcsundef{xspace}%
{%
  \expandafter\edef\csname#1\endcsname{%
    \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
  }%
}%
{%
  \expandafter\edef\csname#1\endcsname{%
    \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
      \noexpand\gls{#1}\noexpand\xspace}%
    }%
  }%
}
}
\newcommand{\newacronym}[4][ ]{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
  \renewcommand{\gls@newacronym}[4]{%
    \PackageError{glossaries}{incompatible acronym style:
      rollback may be required}%
    {something has gone wrong with \string\newacronym. Try
      \string\usepackage{glossaries}[=v4.49]}%
  }%
  \setacronymstyle{long-short}%
  \newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[ ]}%
}
\newcommand{\acrfullformat}[2]{#1\glspace{#2}}
\newrobustcmd{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%

```

```

\acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*\ACRfullplfmt}[3]{%
\acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*\acrfull{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
\new@ifnextchar[{\@acrfull{#1}{#2}}%
{\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
\acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\Acrfull{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][{}]{%
\new@ifnextchar[{\@Acrfull{#1}{#2}}%
{\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
\Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\ACRfull{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][{}]{%
\new@ifnextchar[{\@ACRfull{#1}{#2}}%
{\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
\ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\acrfullpl{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
\new@ifnextchar[{\@acrfullpl{#1}{#2}}%
{\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
\acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][{}]{%
\new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
{\@Acrfullpl{#1}{#2}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
\Acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][{}]{%
\new@ifnextchar[{\@ACRfullpl{#1}{#2}}%

```

```

        {\@ACRfullpl{#1}{#2}[]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymstok{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\glspluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
}
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}%
}
\renewcommand*{\ACRfullfmt}[3]{%

```

```

\glslink[##1]{##2}{-%
  \glsuppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\acrfullplfmt}[3]{-%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{-%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{-%
  \glslink[##1]{##2}{-%
    \glsuppercase{\genplacrfullformat{##2}{##3}}}%
  \renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
  \renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
  \renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
  \renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}
\newcommand*{\acronymsort}[2]{##1}
\newcommand*{\setacronymstyle}[1]{-%
  \ifcsundef{@glsacr@dispstyle@##1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘##1’}{}%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{\acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{##1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\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}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
  \ifglshaslong{\glslabel}{\glsngenacfmt}{\glsngenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\glsacspace{##1}%
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%

```

```

\glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newcommand*\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglschaslong{\glslabel}{\glsacacfm}{\glsacacentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%

```

```

\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
}

```

```

\renewcommand*\acronymentry}[1]{%
  \glentrylong{##1}\glsacspace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort}[2]{##2}%
  \renewcommand*\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
  \newacronymstyle{dua}%
  {%
    \ifdefempty\glscustomtext
    {%
      \ifglshaslong{\glslabel}%
      {%
        \glcifplural
        {%
          \glscapscase
          {%
            \glentrylongpl{\glslabel}\glsinsert
          }%
          {%
            \Glsentrylongpl{\glslabel}\glsinsert
          }%
          {%
            \glsuppercase
            {\glentrylongpl{\glslabel}\glsinsert}%
          }%
        }%
      }%
      {%
        \glscapscase
        {%
          \glentrylong{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylong{\glslabel}\glsinsert
        }%
        {%
          \glsuppercase
          {\glentrylong{\glslabel}\glsinsert}%
        }%
      }%
    }%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

```

```

\renewcommand*\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\glsentrylong{##2}##3\space
      (\acronymfont{\glsentryshort{##2}})}%
\renewcommand*\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsuppercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}%
\renewcommand*\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*\GenericAcronymFields}{}%
  \renewcommand*\acronymentry}[1]{\glsentrylong{##1}}%
  \renewcommand*\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
  \ifglsashaslong{\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}{%
\glsupercase{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glsupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
(\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
\acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
\acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
\acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%

```

```

\renewcommand*\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\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{\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]{\glsgenentryfmt}%
}
\newcommand*\acrfootnote}[3]{\acrlinkfootnote{##1}{##2}{##3}}
\newcommand*\acrlinkfootnote}[3]{%
  \footnote{\glslink{##1}{##2}{##3}}%
}
\newcommand*\acrnoflinkfootnote}[3]{%
  \footnote{##3}%
}
\newrobustcmd*\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{##1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (##1)}%
  \fi
}%
}
\newcommand*\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.

```

```

    Use \string\setacronymstyle{#2} with just glossaries.sty or
    use \string\setabbreviationstyle[acronym]{#3} with
    glossaries-extra.sty}%
}
\RequirePackage{glossary-hypernav}
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserI}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userI\endcsname
}
\newcommand*{\showglouserII}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userII\endcsname
}
\newcommand*{\showglouserIII}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userIII\endcsname
}
\newcommand*{\showglouserIV}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userIV\endcsname
}
\newcommand*{\showglouserV}[1]{%

```

```

    \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
    \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*{\showgloloclist}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
    \csshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
    \show@\glsacronymlists
}
\newcommand*{\showglossaries}{%
    \show@\glo@types
}
\newcommand*{\showglossaryin}[1]{%
    \expandafter\show\csname @glo@type@#1@in\endcsname
}
}

```

```
\newcommand*\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}
```

Change History

1.01 – 2007-05-17	
General: Added range facility in format key	130
\writeist: Added spaces after \delimN and \delimR in ist file	184
1.04 – 2007-08-03	
General: Added \glstextformat	117
1.05 – 2007-08-10	
\glossarysection: added	
\@mkboth to	
\glossarysection	55
\gls@defglossaryentry:	
Changed the default value of the sort key to just the value of the name key	99
1.07 – 2007-09-13	
\@gls@link: fixed bug caused by \theglsentrycounter setting the page number too soon	127
\glsadd: fixed bug caused by \theglsentrycounter setting the page number too soon	181
1.08 – 2007-10-13	
General: Added babel support	48
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle	294
listgroup: changed listgroup style to use \glsgetgrouptitle	293
1.1 – 2008-02-22	
General: msgen now loaded (\new@ifnextchar needed)	6
\@glossarysection: numbered sections and auto label added	56
\@gls@tmpb: changed \toksdef to \newtoks	132
\@gls@toc: numberline added	57
\@p@glossarysection: numbered sections and auto label added	56
numberedsection:	
numberedsection package option added	10
numberline: numberline option added	10
\setglossarysection: new	56
translate: translate option added	30
1.12 – 2008-03-08	
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget)	140
\@GLSpl: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	145
\@Glspl0: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	144
\@glspl0: now uses \glsentrydescplural and \glsentrysymbolplural instead of \glsentrydesc and \glsentrysymbol	143
descriptionplural: new	79
\gls@defglossaryentry:	
Changed default first plural to be first key with s appended (was text key with s appended)	99
descriptionplural support added	99
symbolplural support added	99
\Glsentrydescplural: New	173
\glsentrydescplural: New	173
\Glsentrysymbolplural: New	174
\glsentrysymbolplural: New	174
symbolplural: new	80
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	147–155
\ACRfullpl: new	259
\Acrfullpl: new	259
\acrfullpl: new	259
\acrpluralsuffix: New	256
\gls@defglossaryentry:	
Changed default first value	99

Changed default firstplural value	99	\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	144
Removed restriction on only using \newglossaryentry in the preamble	104	\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	141
\newacronym: Removed restriction on only using \newacronym in the preamble	255	\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used	146
1.14 – 2008-06-17		\@glspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	143
General: added nonumberlist key to \printglossary	242	\@gls@defglossaryentry: Changed def to let	99
added numberedsection key to \printglossary	241	1.17 – 2008-12-26	
\@gls@hypergroup: new	288	General: added xindy support	33
\firstacronymfont: new	260	\@do@esc@wrglossary: new	210
\glsautoprefix: new	10	\@do@seeglossary: new	214
\glsnavhyperlink: changed \edef to \protected@edef	285	\@glo@storeentry: new	105
\glsnavhypertarget: added write to aux file	287	\@gls@glossary: changed definition to use \index instead of \@index	206
\glsnavigation: changed to only use labels for groups that are present	288	\@glsdefaultplural: new	84
1.15 – 2008-08-15		\@glsdefaultsort: new	85
\@gls@link: added \glslabel	127	\@gls@hypernumber: new	252
\@gls@defglossaryentry: check for \@glo@first in description	103	\@glsnoname: new	84
check for \@glo@text in symbol	103	\@glsnonextpages: new	243
\@gls@hypergroup@rerun: new	287	\@gls@defglossaryentry: added nonumberlist key	99
\@glsnavhypertarget: added check if rerun required	287	added parent key	99
\@glssettoctitle: new	48	added see key	99
\printglossary: changed the way the TOC title is set	217	Stored main part of entry format when entry is defined	104
1.16 – 2008-08-27		\@gls@suffixF: new	53
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	143	\@gls@suffixFF: new	53
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used	145	\@gls@wrglossary: modified to allow for xindy support	206
\@Gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	142	\@gls@hyperlink: new	180
		\@gls@hypernumber: modified to allow material to be attached to location	252

<code>\glsnavhyperlink</code> : replaced	1.46 – 2021-09-20
<code>\hyperlink</code> to <code>\@glslink</code>	285
<code>\glsnavhypertarget</code> : replaced	General: changed <code>\edef</code> to
<code>\hypertarget</code> with	<code>\protected@edef</code> 387
<code>\@glsstarget</code>	<code>\@printglossary</code> : changed <code>\xdef</code>
<code>\glssee</code> : new	to <code>\protected@xdef</code> 219
<code>\glsseeformat</code> : new	<code>\gls@defglossaryentry</code> : changed
<code>\glsSetSuffixF</code> : new	<code>\xdef</code> to
<code>\glsSetSuffixFF</code> : new	<code>\protected@xdef</code> 100, 101, 103
<code>\ifglsxindy</code> : new	<code>\glsaddprotectedpagefmt</code> :
<code>\istfilename</code> : added xindy	changed <code>\eappto</code> to
support	<code>\protected@eappto</code> 208
<code>\newglossarystyle</code> : made	2.01 – 2009 May 30
<code>\newglossarystyle long</code>	<code>\@gls@link</code> : moved
<code>nonumberlist</code> : new	<code>\@do@wrglossary</code> before term
<code>\nopostdesc</code> : new	is displayed to prevent
<code>order</code> : order package option	unwanted whatsit 128
added	<code>\forallglossaries</code> : replaced
<code>parent</code> : new	<code>\ifthenelse</code> with <code>\ifx</code> 67
<code>\printglossary</code> : added check to	<code>\forlsgentries</code> : replaced
determine if <code>\printglossary</code>	<code>\ifthenelse</code> with <code>\ifx</code> 67
is already defined	<code>\glsdefmain</code> : new 19
added print language to aux	<code>\glsdescwidth</code> : changed
file	<code>\linewidth</code> to <code>\hsize</code> . 296, 318
<code>see</code> : new	<code>\glslistdottedwidth</code> : changed
<code>\writeist</code> : added xindy support 184	<code>\linewidth</code> to <code>\hsize</code> 295
1.18 – 2009-01-14	<code>\glspagelistwidth</code> : changed
<code>\@gls@loadlist</code> : new	<code>\linewidth</code> to <code>\hsize</code> . 296, 318
<code>\@gls@loadlong</code> : new	<code>nomain</code> : added <code>nomain</code> package
<code>\@gls@loadsuper</code> : new	option 20
<code>\@gls@loadtree</code> : new	<code>\writeist</code> : removed <code>item_02</code> -
<code>\gls@defglossaryentry</code> :	no such <code>makeindex</code> key 189
Changed default value of sort	2.02 – 2007-07-13
to <code>\@glsdefaultsort</code>	<code>\@printglossary</code> : suppressed
moved sort sanitization to	warning globally rather than
<code>\newglossaryentry</code>	locally 220
<code>\glsstarget</code> : new	2.02 – 2009-07-13
<code>nolist</code> : new	<code>\glossarysection</code> : changed
<code>nolong</code> : new	<code>\@mkboth</code> to <code>\glossarymark</code> 55
<code>nostyles</code> : new	<code>\gls glossarymark</code> : New 55
<code>nosuper</code> : new	2.03 – 2009-09-23
<code>notree</code> : new	<code>\@GLS@</code> : Added check for
<code>\oldacronym</code> : new	<code>hyperfirst</code> 143
<code>sort</code> : moved sanitization to	<code>\@GLSp1</code> : Added check for
<code>\newglossaryentry</code>	<code>hyperfirst</code> 145
1.19 – 2009-03-02	<code>\@GLS@</code> : Added check for
<code>\glsclearpage</code> : new	<code>hyperfirst</code> 142
<code>\glsdisp</code> : new	<code>\@GLSp1@</code> : Added check for
	<code>hyperfirst</code> 144
	<code>\@gls@</code> : Added check for
	<code>hyperfirst</code> 141

<code>\@gls@link</code> : new	126	<code>\gls@defglossaryentry</code> : added	
<code>\@gls@link</code> : added <code>\leavevmode</code>	127	user1-6 keys	99
Moved entry existence check to		<code>\glsadd</code> : fixed bug that ignored	
avoid duplicate code	127	counter	181
<code>\@glsdisp</code> : Added check for		<code>\Glsentryuseri</code> : new	176
<code>hyperfirst</code>	146	<code>\glsentryuseri</code> : new	176
<code>\@glspl@</code> : Added check for		<code>\Glsentryuserii</code> : new	176
<code>hyperfirst</code>	143	<code>\glsentryuserii</code> : new	176
<code>\gls glossarymark</code> : Added check		<code>\Glsentryuseriii</code> : new	176
to see if it's already defined	55	<code>\glsentryuseriii</code> : new	176
<code>hyperfirst</code> : new	32	<code>\Glsentryuseriv</code> : new	177
2.04 – 2009-11-10		<code>\glsentryuseriv</code> : new	177
<code>\@GLS@</code> : Changed test to check if		<code>\Glsentryuserv</code> : new	177
glossary type has been		<code>\glsentryuserv</code> : new	177
identified as a list of		<code>\Glsentryuservi</code> : new	177
acronyms	143	<code>\glsentryuservi</code> : new	177
<code>\@GLSpl@</code> : Changed test to check if		<code>\ns@newglossary</code> : added check to	
glossary type has been		determine if	
identified as a list of		<code>\gls@<i>type</i>@display</code> and	
acronyms	145	<code>\gls@<i>type</i>@displayfirst</code>	
<code>\@Gls@</code> : Changed test to check if		have been defined.	76
glossary type has been		<code>\SetAcronymLists</code> : new	22
identified as a list of		<code>\SetDefaultAcronymDisplayStyle</code> :	
acronyms	142	new	275
<code>\@Glspl@</code> : Changed test to check		<code>\SetDefaultAcronymStyle</code> : new	275
if glossary type has been		<code>\SetDescriptionAcronymDisplayStyle</code> :	
identified as a list of		new	276
acronyms	144	<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>\@glossaryentryfield</code> : new	105	new	276
<code>\@glossarysubentryfield</code> : new	105	<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\@gls@</code> : Changed test to check if		new	276
glossary type has been		<code>\SetDUADisplayStyle</code> : new	277
identified as a list of		<code>\SetFootnoteAcronymDisplayStyle</code> :	
acronyms	141	new	276
<code>\@glsacronymlists</code> : new	21	<code>\SetSmallAcronymDisplayStyle</code> :	
<code>\@glsdisp</code> : Changed test to check		new	277
if glossary type has been		2.05 – 2010-02-06	
identified as a list of		<code>\@glsdisp</code> : Added closing brace.	
acronyms	146	Patch provided by Sergiu	
<code>\@glspl@</code> : Changed test to check		Dotenco	146
if glossary type has been		Removed spurious brace. Patch	
identified as a list of		provided by Sergiu Dotenco	146
acronyms	143	<code>\writeist</code> : Added <code>\string</code>	
<code>\@newglossaryentryposthook</code> :		before opening and closing	
new	104	braces. Patch provided by	
<code>\@newglossaryentryprehook</code> :		Sergiu Dotenco	189
new	104	2.06 – 2010-06-14	
<code>acronymlists</code> : new	22	<code>\altnewglossary</code> : new	77
<code>\DeclareAcronymList</code> : new	21	<code>\CustomAcronymFields</code> : new	278
<code>\DefineAcronymSynonyms</code> : new	273	<code>\CustomNewAcronymDef</code> : new	278

\SetCustomDisplayStyle: new .	278	added \glsnextpages	219
\SetCustomStyle: new	278	make toctitle default to title .	219
2.07 – 2010-07-10		\@xdyattributelist: new	58
General: glsadd format key stored		\acrfootnote: new	275
in \@glsnumberformat (was		\ACRfull: added starred version	259
mistakenly stored in		\Acrfull: added starred version	258
\@glo@format)	181	\acrfull: added starred version	258
3.0 – 2010-07-12		\ACRfullpl: added starred	
\@makeglossary: Added check for		version	259
savewrites	194	\Acrfullpl: added starred	
\@gls@wrglossary: modified to		version	259
take into account savewrites	206	\acrfullpl: added starred	
3.0 – 2010/03/31		version	259
\@set@glo@numformat: added 4th		\acrlinkfootnote: new	275
argument	130	\acrnolinkfootnote: new	275
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	39
hyperlink	253	counter: replaced \@ifundefined	
etoolbox now loaded	6	with \ifcsundef	80
replaced \@ifundefined with		counterwithin: new	14
\ifcsundef	46, 49, 123, 241	entrycounter: new	14
\@do@esc@wrglossary: added		glossaryentry: new	14
check for hyper location		\glossarysection: replaced	
prefix	212	\@ifundefined with	
modified to use new format .	210	\ifcsundef	55
\@glossarysec: replaced		\glossarystyle: replaced	
\@ifundefined with		\@ifundefined with	
\ifcsundef	10	\ifcsundef	251
\@do@seeglossary: Sanitize and		glossarysubentry: new	15
escape cross-referencing		\gls@codepage: replaced	
information	214	\@ifundefined with	
\@gls@counterwithin: new	15	\ifcsundef	34
\@gls@ifinlist: new	58	\gls@defglossaryentry: added	
\@gls@link: added		\@gls@defsort	103
\@gls@saveentrycounter	128	added short and long keys	100
added \@gls@setsort	128	replaced \@ifundefined with	
\@gls@saveentrycounter: new	128	\ifcsundef	100
\@gls@setupsort@def: new	17	\gls@doclearpage: replaced	
\@gls@setupsort@standard: new	16	\@ifundefined with	
\@gls@setupsort@use: new	17	\ifcsundef	57
\@gls@xdy@locationlist: new	62	\glsadd: added	
\@glslink: replaced		\@gls@saveentrycounter	181
\@ifundefined with		\GlsAddXdyCounters: new	59
\ifcsundef	139	\glsentrycounterlabel: new	245
\@glsnextpages: new	243	\glsentryitem: new	245
\@print@glossary: replaced		\Glsentrylong: new	178
\@ifundefined with		\glsentrylong: new	178
\ifcsundef	220	\Glsentrylongpl: new	178
\@printglossary: added		\glsentrylongpl: new	178
\currentglossary	219	\Glsentryshort: new	178

<code>\glentryshort</code> : new	177	replaced <code>\@ifundefined</code> with	
<code>\Glentryshortpl</code> : new	178	<code>\ifcsundef</code>	76
<code>\glentryshortpl</code> : new	178	<code>\oldacronym</code> : replaced	
<code>\glsgrouptitle</code> : replaced		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	255
<code>\ifcsundef</code>	249	<code>\printglossary</code> : replaced	
<code>\glsglossarymark</code> : replaced		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	217
<code>\ifcsundef</code>	55	<code>savewrites</code> : new	39
<code>\glshyperlink</code> : changed default		<code>see</code> : added	
from <code>\glentryname</code> to		<code>\@glo@seeautonumberlist</code>	81
<code>\glentrytext</code>	180	<code>seeautonumberlist</code> : new	12
<code>\glshyphnumber</code> : replaced		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\@ifundefined</code> with		expanded options link	
<code>\ifcsundef</code>	252	options	276
<code>\glnumberformat</code> : replaced		<code>\setentrycounter</code> : added	
<code>\@ifundefined</code> with		optional argument	250
<code>\ifcsundef</code>	53	<code>short</code> : new	83
<code>\glhrefentry</code> : new	244	<code>shortplural</code> : new	84
<code>\glresetsubentrycounter</code> :		<code>\showacronymlists</code> : new	283
new	244	<code>\showglocounter</code> : new	280
<code>\glseeitem</code> : hyperlink uses		<code>\showgloDESC</code> : new	281
<code>\glseeitemformat</code> instead of		<code>\showgloDESCplural</code> : new	281
<code>\glentryname</code>	216	<code>\showglofirst</code> : new	279
<code>\glseeitemformat</code> : new	216	<code>\showglofirstpl</code> : new	279
<code>\glssortnumberfmt</code> : new	16	<code>\showgloflag</code> : new	283
<code>\glststepentry</code> : new	244	<code>\showgloindex</code> : new	282
<code>\glststepsubentry</code> : new	244	<code>\showglolevel</code> : new	279
<code>\glssubentrycounterlabel</code> :		<code>\showglongame</code> : new	281
new	245	<code>\showgloparent</code> : new	279
<code>\glssubentryitem</code> : new	245	<code>\showgloplural</code> : new	279
<code>\ifglossaryexists</code> : replaced		<code>\showglosort</code> : new	282
<code>\@ifundefined</code> with		<code>\showglossaries</code> : new	283
<code>\ifcsundef</code>	68	<code>\showglossarycounter</code> : new	284
<code>\ifglentryexists</code> : replaced		<code>\showglossaryentries</code> : new	284
<code>\@ifundefined</code> with		<code>\showglossaryin</code> : new	283
<code>\ifcsundef</code>	69	<code>\showglossaryout</code> : new	284
<code>\istfile</code> : deprecated	204	<code>\showglossarytitle</code> : new	284
<code>long</code> : new	84	<code>\showglosymbol</code> : new	282
<code>longplural</code> : new	84	<code>\showglosymbolplural</code> : new	282
<code>\newglossaryentry</code> : replaced		<code>\showglotext</code> : new	279
<code>\DeclareRobustCommand</code> with		<code>\showglotype</code> : new	280
<code>\newrobustcmd</code>	87	<code>\showglouserI</code> : new	280
<code>\newglossarystyle</code> : replaced		<code>\showglouserII</code> : new	280
<code>\@ifundefined</code> with		<code>\showglouserIII</code> : new	280
<code>\ifcsundef</code>	251	<code>\showglouserIV</code> : new	281
<code>nonumberlist</code> : now boolean	82	<code>\showglouserV</code> : new	281
<code>\ns@newglossary</code> : added		<code>\showglouserVI</code> : new	281
<code>\@gls@defsortcount</code>	77	<code>sort</code> : new	15
		<code>subentrycounter</code> : new	15

<code>\gls@defglossaryentry</code> : added numberlist element	103	<code>\glspostinline</code> : replaced “.” with <code>\glspostdescription</code>	291
<code>\gls@save@numberlist</code> : new . .	216	index: added check for <code>glsnogroupskip</code>	334
<code>\gls@wrglossary</code> : added check for glossary file defined . . .	206	list: added check for <code>glsnogroupskip</code>	293
<code>\glsdisplaynumberlist</code> : new .	179	long: added check for <code>glsnogroupskip</code>	296
<code>\glsentrycounter</code> : set default value	128	long3col: added check for <code>glsnogroupskip</code>	298
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	179	long4col: added check for <code>glsnogroupskip</code>	299
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>) . .	179	longragged: added check for <code>glsnogroupskip</code>	308
<code>\glsentrynumberlist</code> : new . . .	179	longragged3col: added check for <code>glsnogroupskip</code>	309
<code>\glsmoveentry</code> : new	105	nogroupskip: new	14
<code>\glsresetentrycounter</code> : new .	244	nopostdot: new	13
<code>\ifglshaschildren</code> : new	70	super: added check for <code>glsnogroupskip</code>	319
<code>\ifglshasparent</code> : new	71	super3col: added check for <code>glsnogroupskip</code>	321
indexonlyfirst: new	32	super4col: added check for <code>glsnogroupskip</code>	323
<code>\makeglossaries</code> : added list parser	199	superragged: added check for <code>glsnogroupskip</code>	326
<code>\renewglossarystyle</code> : new . . .	251	superragged3col: added check for <code>glsnogroupskip</code>	328
savenumberlist: new	12	tree: added check for <code>glsnogroupskip</code>	335
<code>\showglossaryentries</code> : fixed misspelt command	284	treenoname: added check for <code>glsnogroupskip</code>	337
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	277		
ucmark: new	14		
3.03 – 2012/09/21			
General: allow title to set toctitle	241		
<code>\@gls@sanitizesort</code> : new	25	3.04 – 2012-11-11	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	16	<code>altlist</code> : replaced <code>\newline</code> with paragraph break	294
<code>\@printglossary</code> : allow title to override default toctitle . . .	218	3.04 – 2012-11-18	
<code>altlongragged4col</code> : added check for <code>glsnogroupskip</code>	311	General: Added check for doc package	6
<code>altsuperragged4col</code> : added check for <code>glsnogroupskip</code> . .	330	added datatool-base as a required package	51
<code>almtree</code> : added check for <code>glsnogroupskip</code>	340	added local key	124
<code>\glsinlinedescformat</code> : new . .	291	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	213
<code>\glsinlineemptydescformat</code> : new	291	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	211
<code>\glsinlinenameformat</code> : new . .	291	<code>\@gls@esc@sdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphpage</code> ,	
<code>\glsinlinepostchild</code> : new . . .	291		
<code>\glsinlinesubdescformat</code> : new	291		
<code>\glsinlinesubnameformat</code> : new	291		

\gls@Alphpage and			
\gls@romanpage	131		
\@print@glossary: Moved aux			
write to end of document to			
prevent unwanted whatsit			
occurring here.	220		
\gls@Alphpage: new	208		
\gls@alphpage: new	207		
\gls@disablepagerefexpansion:			
new	207		
\gls@numberpage: new	208		
\gls@protected@pagefmts: new	207		
\gls@romanpage: new	208		
\glsdefmain: added check for doc			
package	19		
\glsorg@endtheglossary: new	6		
\glsorg@theglossary: new	6		
\PrintChanges: new	6		
3.05 – 2013-04-21			
\@do@esc@wrglossary: add			
Roman case. Fixed bugs in			
the else statements	211		
\@gls@link: added check for			
“nohypertypes”	127		
\gls@protected@pagefmts:			
added Roman to list	207		
\gls@Romanpage: new	208		
\glsgetgrouplabel: fixed bug			
(typo in \equal)	250		
\glsmcols: new	313		
mcolalttree: replaced ‘2’ with			
\glsmcols	317		
mcolindex: replaced ‘2’ with			
\glsmcols	313		
mcolindexspannav: replaced ‘2’			
with \glsmcols	314		
mcoltree: replaced ‘2’ with			
\glsmcols	314		
mcoltreenoname: replaced ‘2’			
with \glsmcols	315		
mcoltreesspannav: replaced ‘2’			
with \glsmcols	315		
\nopostdesc: made robust	51		
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	34		
\findrootlanguage: Obsoleted	65		
3.07 – 2013-07-05			
\@gls@link: fixed bug that failed			
to find entry in list	127		
\glossarypreamble: modified to			
work with			
\setglossarypreamble	54		
\gls@docclearpage: added check			
for openright	57		
\glspostdescription: Added			
spacefactor code	13		
\GlsSetXdyCodePage: Added			
check for fontspec	66		
\SetDescriptionAcronymDisplayStyle:			
now using			
\glsdoparenifnotempty	276		
\setglossarypreamble: new	54		
3.08a – 2013-08-30			
altlist: updated altlist style to			
use \glossentry and			
\subglossentry	294		
inline: updated inline style to			
use \glossentry and			
\subglossentry	289		
list: updated list style to use			
\glossentry and			
\subglossentry	293		
listdotted: updated listdotted			
style to use \glossentry and			
\subglossentry	295		
3.08a – 2013-09-28			
General: added nogroupskip key			
to \printglossary	242		
removed definition of			
\@glossaryentryfield	376		
removed definition of			
\@glossarysubentryfield	376		
\@glo@storeentry: no longer			
need to check for special			
characters in any of the fields			
other than sort	106		
updated for \glossentry	106		
\@glossaryentryfield: switched			
to \glossentry	105		
\@glossarysubentryfield:			
switched to \subglossentry	105		
\@gls@sanitizedesc: deprecated	24		
altlongragged4col: updated to			
use \glossentry and			
\subglossentry	310		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	338	\subglossentry	326
\compatibleglossentry: new .	246	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	247	\subglossentry	335
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	247	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	251	new	25
\glossentry: new	246	\@gls@default@value: new ...	80
\Glossentrydesc: new	247	\Glsentrydesc: made robust ..	173
\glossentrydesc: new	247	\Glsentrydescplural: made	
\Glossentryname: new	246	robust	173
\glossentryname: new	246	\Glsentryfirst: made robust ..	174
\Glossentrysymbol: new	247	\Glsentryfirstplural: made	
\glossentrysymbol: new	247	robust	175
\gls@assign@desc@field: new .	24	\Glsentryfull: made robust ..	179
\gls@assign@descplural@field:		\Glsentryfullpl: made robust	179
new	25	\Glsentrylong: made robust ..	178
\gls@assign@field: new	86	\Glsentrylongpl: made robust	178
\gls@ifnotmeasuring: new ...	107	\Glsentryname: made robust ..	171
\glsaddallunused: new	182	\Glsentryplural: made robust	174
\glsexpandfields: new	86	\Glsentryshort: made robust .	178
\glsnoexpandfields: new	87	\Glsentryshortpl: made robust	178
\glssee: made robust	215	\Glsentrysymbol: made robust	174
\glsseeformat: made robust ..	215	\Glsentrysymbolplural: made	
\glsseeitem: made robust	216	robust	174
\glsseelist: made robust	215	\Glsentrytext: made robust ..	173
\ifglsdescsuppressed: new ..	71	\Glsentryuseri: made robust .	176
\ifglsdesc: new	71	\Glsentryuserii: made robust	176
\ifglsymbol: new	71	\Glsentryuseriii: made robust	176
index: added paragraph break at		\Glsentryuseriv: made robust	177
end of environment	333	\Glsentryuserv: made robust .	177
updated to use \glossentry		\Glsentryuservi: made robust	177
and \subglossentry	333	\glstextup: new	256
long: updated to use		\ifglsymbol: changed test	
\glossentry and		to check for	
\subglossentry	296	\@gls@default@symbol	71
longragged: updated to use		3.10a – 2013-09-28	
\glossentry and		\gls@assign@type@field: new .	24
\subglossentry	307	3.10a – 2013-10-13	
longragged3col: updated to use		\@gls@keymap: new	90
\glossentry and		\@gls@provide@newglossary:	
\subglossentry	309	new	74
\setglossarystyle: new	250	\@gls@writedef: new	89
\setglossentrycompatibility:		\@glsdefaultplural: Obsolete .	84
new	247	\@glsnodesc: new	84
\subglossentry: new	248	\@print@glossary: Added	
		providecommand code to aux	
		file	221

<code>\gls@defglossaryentry:</code>		changed to just use	
Changed to using		<code>\glsentrytext</code>	147
<code>\@gls@default@value</code>	99	changed to just use	
new	98	<code>\Glsentryuseriii</code>	158
<code>\gls@writedefhook:</code> new	98	changed to just use	
<code>\makeglossaries:</code> Added		<code>\glsentryuseriii</code>	158
providecommand code to aux		changed to just use	
file	198	<code>\Glsentryuserii</code>	157
<code>\new@glossaryentry:</code> new	88	changed to just use	
<code>\ns@newglossary:</code> added		<code>\glsentryuserii</code>	157
<code>\@gls@provide@newglossary</code>	76	changed to just use	
3.11a – 2013-10-15		<code>\Glsentryuseriv</code>	159
General: added <code>\glslabel</code> ,		changed to just use	
<code>\glsifplural</code> , <code>\glscapscase</code> ,		<code>\glsentryuseriv</code>	159
<code>\glsinsert</code> and		changed to just use	
<code>\glscustomtext</code>	162–169	<code>\Glsentryuseri</code>	156
changed to just use		changed to just use	
<code>\Glsentrydescplural</code>	153	<code>\glsentryuseri</code>	156
changed to just use		changed to just use	
<code>\glsentrydescplural</code>	153	<code>\Glsentryuservi</code>	161
changed to just use		changed to just use	
<code>\Glsentrydesc</code>	152	<code>\glsentryuservi</code>	161
changed to just use		changed to just use	
<code>\glsentrydesc</code>	152	<code>\Glsentryuserv</code>	160
changed to just use		changed to just use	
<code>\Glsentryfirstplural</code>	150	<code>\glsentryuserv</code>	160
changed to just use		Now requires <code>textcase</code>	5
<code>\glsentryfirstplural</code>	150	<code>\@ACRlong:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentryfirst</code>	148	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	371
<code>\glsentryfirst</code>	148	<code>\@ACRshort:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentryname</code>	151	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	369
<code>\glsentryname</code>	151	<code>\@Acrlong:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentryplural</code>	149	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	370
<code>\glsentryplural</code>	149	<code>\@Acrshort:</code> added <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentrysymbolplural</code>	155	<code>\glsinsert</code> and	
changed to just use		<code>\glscustomtext</code>	369
<code>\glsentrysymbolplural</code>	155	<code>\@GLS@:</code> add <code>\glslabel</code> ,	
changed to just use		<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\Glsentrysymbol</code>	154	<code>\glscustomtext</code> and	
changed to just use		<code>\glsinsert</code>	143
<code>\glsentrysymbol</code>	154	change to using <code>\glsentryfmt</code>	
Changed to just use		style commands	143
<code>\Glsentrytext</code>	147		

removed \MakeUppercase (now moved to \glentryfmt) ..	143	\@glspl@: add \glslabel, \glsifplural, \glscapscase, \glsinsert	143
\@GLSpl: add \glslabel, \glsifplural, \glscapscase, \glsinsert	145	change to using \glentryfmt style commands	145
removed \MakeUppercase as now dealt with in \glentryfmt	145	acronymlists: replaced \@addtoacronymlists with \DeclareAcronymList	22
\@Gls@: add \glsifplural, \glscapscase, \glsinsert	142	compatible-3.07: new	39
change to using \glentryfmt style commands	142	\defglsdisplay: obsolete	123
removed \makefirstuc (now dealt with in \glentryfmt)	142	\defglsdisplayfirst: obsolete	123
\@Glspl@: add \glsifplural, \glscapscase, \glsinsert	144	\defglentryfmt: new	75
change to using \glentryfmt style commands	144	\forglentries: replaced \ifx with \ifdefempty	67
removed \makefirstuc (now dealt with in \glentryfmt)	144	\gls@assign@desc: new	98
\@acrlong: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glsinsert	370	\gls@defglossaryentry: Fixed default counter if none supplied	102
\@acrshort: added \glslabel, \glsifplural, \glscapscase, \glsinsert and \glsinsert	369	\gls@doentryfmt: new	75
\@gls@: add \glslabel, \glsifplural, \glscapscase, \glsinsert	141	\glsdisplay: obsolete	123
change to using \glentryfmt style commands	141	\glsdisplayfirst: obsolete ..	123
\@gls@noexpand@fields: Fixed bug expand replaced with noexpand	85	\glsentryfmt: new	118
\@glsdisp: add \glslabel, \glsifplural, \glscapscase, \glsinsert	146	\glsgetgrouptitle: Added check in case non-Latin alphabet in use	249
change to using \glentryfmt style commands	146	\gls glossarymark: replaced \MakeUppercase with \mfirstucMakeUppercase ..	55
		\glsnavigation: switched to using \@gls@getgrouptitle	288
		\ifglshasdesc: replaced \ifdefempty with \ifcempty	71
		\ifglshaslong: new	72
		\ifglshasshort: new	72
		\ifglshassymbol: replaced \ifdefempty with \ifcempty	71
		\ifglsused: replaced \ifthenelse with \ifbool ..	69
		\longnewglossaryentry: new ..	98
		\ns@newglossary: replaced \glsdisplay and \glsdisplayfirst with \glentryfmt	76
		numbers: new	41
		\SetCustomDisplayStyle: updated to use \defglentryfmt	278

<code>\SetDefaultAcronymDisplayStyle:</code> changed to use <code>\defglentryfmt</code> 275	<code>\gls@assign@desc@field:</code> changed to use <code>\glssetnoexpandfield</code> 24
<code>\SetDescriptionAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 276	<code>\gls@assign@descplural@field:</code> changed to use <code>\glssetnoexpandfield</code> 25
<code>\SetDescriptionDUAAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 276	<code>\gls@assign@name@field:</code> changed to use <code>\glssetnoexpandfield</code> 25
<code>\SetDescriptionFootnoteAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 276	<code>\gls@assign@type@field:</code> changed to use <code>\glssetexpandfield</code> 24
<code>\SetDUADisplayStyle:</code> updated to use <code>\defglentryfmt</code> . . 277	<code>\gls@checkseeallowed:</code> new . . 81 <code>\glsaddallunused:</code> set default to <code>\@glo@types</code> 182
<code>\SetFootnoteAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 276	<code>\Glsentryfull:</code> changed to use <code>\acrfullformat</code> 179
<code>\SetSmallAcronymDisplayStyle:</code> updated to use <code>\defglentryfmt</code> 277	<code>\glsentryfull:</code> changed to use <code>\acrfullformat</code> 178
<code>\setupglossaries:</code> new 45	<code>\Glsentryfullpl:</code> changed to use <code>\acrfullformat</code> 179
<code>\showglolong:</code> new 282	<code>\glsentryfullpl:</code> changed to use <code>\acrfullformat</code> 179
<code>\showgloshort:</code> new 282	<code>\gls glossarymark:</code> renamed <code>\glossarymark</code> to <code>\gls glossarymark</code> to avoid conflict with memoir 55
<code>symbols:</code> new 41	<code>\glsprestandardsort:</code> new . . . 15
3.12a – 2013-10-16 <code>\gls@defglossaryentry:</code> added <code>\glslabel</code> 99	<code>\glssetexpandfield:</code> new 24
<code>\glsaddkey:</code> new 92	<code>\glssetnoexpandfield:</code> new . . 24
3.13a – 2013-11-05 <code>\@gls@assign@symbol@field:</code> changed to use <code>\glssetnoexpandfield</code> 25	long: switched to <code>\tabularnewline</code> 296
<code>\@gls@assign@symbolplural@field:</code> changed to use <code>\glssetnoexpandfield</code> 25	long3col: switched to <code>\tabularnewline</code> 298
<code>\@gls@link:</code> removed <code>\relax</code> . . 128	long3colheader: switched to <code>\tabularnewline</code> 298
<code>\@gls@notranslatorhook:</code> new . . 30	long3colheaderborder: switched to <code>\tabularnewline</code> 299
<code>\@gls@setupsort@standard:</code> moved <code>\@gls@santizesort</code> to <code>\glsprestandardsort</code> 16	long4col: switched to <code>\tabularnewline</code> 299
altsuper4colheader: switched to <code>\tabularnewline</code> 324	long4colheader: switched to <code>\tabularnewline</code> 300
altsuper4colheaderborder: switched to <code>\tabularnewline</code> 325	longheader: switched to <code>\tabularnewline</code> 297
<code>\glossarysection:</code> changed <code>\glossarymark</code> to <code>\gls glossarymark</code> 55	longheaderborder: switched to <code>\tabularnewline</code> 297
	see: added <code>\gls@checkseeallowed</code> 81
	super: switched to <code>\tabularnewline</code> 319

super3col: switched to		\@acrfull: now using	
\tabularnewline	321	\acrfullfmt	258
super3colheader: switched to		\@gls@indexdef: new	42
\tabularnewline	321	\@gls@numbersdef: new	41
super4col: switched to		\@gls@symbolsdef: new	41
\tabularnewline	322	\ACRfullfmt: new	257
super4colheader: switched to		\Acrfullfmt: new	257
\tabularnewline	323	\acrfullfmt: new	257
super4colheaderborder:		\ACRfullplfmt: new	258
switched to		\Acrfullplfmt: new	258
\tabularnewline	324	\acrfullplfmt: new	258
superheader: switched to		\acronymentry: new	262
\tabularnewline	320	dua: new	268
superheaderborder: switched to		dua-desc: new	270
\tabularnewline	320	footnote: new	270
ucmark: added check for memoir	14	footnote-desc: new	272
3.14a – 2013-11-12		footnote-sc: new	272
General: new	343	footnote-sc-desc: new	272
\@glswritefiles: renamed		footnote-sm: new	272
\glswritefiles to		footnote-sm-desc: new	273
\@glswritefiles and used		\Genacrfullformat: new	122
“savewrites” option to set		\genacrfullformat: new	122
\glswritefiles	204	\GenericAcronymFields: new	262
acronyms: new	20	\Genplacrfullformat: new	123
compatible-2.07: added check		\genplacrfullformat: new	122
for 2.07 before setting 3.07		\Glsentryfull: bug fix: added	
compatibility	39	missing \acronymfont	179
\@gls@defglossaryentry: added		\Glsentryfull: bug fix: added	
check for existence of default		missing \acronymfont	178
glossary	100	\Glsentryfullpl: bug fix: added	
set the default for firstplural to		missing \acronymfont	179
be the value of plural	102	\Glsentryfullpl: bug fix: added	
\longprovideglossaryentry:		missing \acronymfont	179
new	98	\glsgenacfmt: new	120
notranslate: new	30	\GlsUseAcrEntryDisplayStyle:	
\provideglossaryentry: new	87	new	263
xindygloss: new	34	\GlsUseAcrStyleDefs: new	263
4.0 – 2013-11-14		index: new	41
\@gls@defglossaryentry: added		long-sc-short: new	265
check for first key	102	long-sc-short-desc: new	267
super: fixed typo in		long-short: new	263
\subglossentry		long-short-desc: new	266
(\glossentrydesc)	319	long-sm-short: new	265
4.01 – 2013-11-16		long-sm-short-desc: new	267
General: fixed non-value options		\newacronymstyle: new	262
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	166–169	sc-short-long: new	266

replaced #1 with		\glslocalreset: added	
\@gls@label	100	\glsdetoklabel	109
replaced \ifthenelse with		\glslocalunset: added	
\ifdefequal	101	\glsdetoklabel	110
\glsadd: added \glsdetoklabel	181	\glsmoveentry: added	
\glsaddkey: switched to using		\glsdetoklabel	105
\@gls@field@link	93	replaced \ifthenelse with	
\glsdetoklabel: new	69	\ifdefequal	105
\glsdisplaynumberlist: added		\glsrefentry: added	
\glsdetoklabel	180	\glsdetoklabel	244
\glsdoifexistsorwarn: new . .	70	\glsreset: added	
\glsentryaccess: switched to		\glsdetoklabel	109
using \@gls@entry@field .	356	\glsseelist: added	
\glsentrydescaccess: switched		\expandafter commands .	215
to using \@gls@entry@field	357	\glsstepentry: added	
\glsentrydescpluralaccess:		\glsdetoklabel	244
switched to using		\glsstepsubentry: added	
\@gls@entry@field	357	\glsdetoklabel	244
\glsentryfirstaccess: switched		\glsunset: added	
to using \@gls@entry@field	356	\glsdetoklabel	109
\glsentryfirstplural: added		\ifglsdescsuppressed: added	
\glsdetoklabel	175	\glsdetoklabel	71
\glsentrylongaccess: switched		fixed typo	71
to using \@gls@entry@field	357	\ifglsentryexists: added	
\glsentrylongpluralaccess:		\glsdetoklabel	69
switched to using		\ifglschaschildren: added	
\@gls@entry@field	357	\glsdetoklabel	70
\glsentrypluralaccess:		\ifglschasdesc: added	
switched to using		\glsdetoklabel	71
\@gls@entry@field	356	\ifglschasfield: new	72
\glsentryshortaccess: switched		\ifglschaslong: added	
to using \@gls@entry@field	357	\glsdetoklabel	72
\glsentryshortpluralaccess:		\ifglschasparent: added	
switched to using		\glsdetoklabel	71
\@gls@entry@field	357	\ifglschasshort: added	
\glsentrysymbolaccess:		\glsdetoklabel	72
switched to using		\ifglschassymbol: added	
\@gls@entry@field	357	\glsdetoklabel	71
\glsentrysymbolpluralaccess:		replaced \ifcsemtyp with	
switched to using		\ifdefempty and replaced	
\@gls@entry@field	357	\ifx with \ifdefequal	71
\glsentrytextaccess: switched		\ifglsused: added	
to using \@gls@entry@field	356	\glsdetoklabel	69
\glsgenacfmt: redefined to use		long-sc-short-desc: redefined	
accessibility information . .	366	to use accessibility	
\glsgenentryfmt: redefined to		information	379
use accessibility information	363	long-short: redefined to use	
\glshyperlink: added		accessibility information . .	377
\glsdetoklabel	180	long-short-desc: redefined to	
		use accessibility information	379

<code>\@gls@noidx@sanitizesort:</code>		<code>\gls@defglossaryentry:</code>	
new	25	replaced <code>\ifx</code> with	
<code>\@gls@nosanitizesort:</code> new	25	<code>\ifdefvoid</code>	104
<code>\@gls@sanitizesort:</code> new	25	<code>\glsnoidxdisplayloc:</code> new	239
<code>\@glo@addchildren:</code> new	228	<code>\glsnoidxdisplayloclisthandler:</code>	
<code>\@glo@do@sortentries:</code> new	228	new	239
<code>\@glo@grabfirst:</code> new	237	<code>\glsnoidxloclist:</code> new	239
<code>\@glo@sortedinsert:</code> new	229	<code>\glsnoidxloclisthandler:</code> new	239
<code>\@glo@sortentries:</code> new	227	<code>\glsnoidxstripaccents:</code> new	26
<code>\@glo@sorthandler@case:</code> new	230	<code>\makeglossaries:</code> Moved	
<code>\@glo@sorthandler@letter:</code>		definition of <code>\glswrite</code> to	
new	229	<code>\makeglossaries</code>	198
<code>\@glo@sorthandler@nocase:</code>		<code>\makenoidxglossaries:</code> new	200
new	230	<code>\printglossary:</code> changed to use	
<code>\@glo@sorthandler@word:</code> new	229	new <code>\@printglossary</code>	217
<code>\@glo@sortmacro@case:</code> new	231	<code>\printnoidxglossaries:</code> new	218
<code>\@glo@sortmacro@def:</code> new	231	<code>\printnoidxglossary:</code> new	218
<code>\@glo@sortmacro@def@do:</code> new	232	<code>\showgloloclist:</code> new	283
<code>\@glo@sortmacro@letter:</code> new	230	<code>\warn@noprintglossary:</code>	
<code>\@glo@sortmacro@nocase:</code> new	231	Activate warning in	
<code>\@glo@sortmacro@standard:</code>		<code>\makeglossaries</code>	216
new	231	<code>\writeist:</code> checked for definition	
<code>\@glo@sortmacro@use:</code> new	232	of <code>\glswrite</code>	184, 189
<code>\@glo@sortmacro@word:</code> new	230	4.06 – 2014-03-12	
<code>\@gls@noidx@do:</code> new	237	General: added	
<code>\@gls@noidx@getgrouptitle:</code>		<code>\glsifhyper</code>	162–169
new	249	<code>\@GLS@:</code> added <code>\glsifhyper</code>	143
<code>\@gls@noref@warn:</code> new	204	<code>\@GLSpl:</code> added <code>\glsifhyper</code>	145
<code>\@gls@reference:</code> new	240	<code>\@Gls@:</code> added <code>\glsifhyper</code>	142
<code>\@gls@warnonglossdefined:</code> new	24	<code>\@Glspl@:</code> added <code>\glsifhyper</code>	144
<code>\@gls@warnontheglossdefined:</code>		<code>\@gls@:</code> added <code>\glsifhyper</code>	141
new	24	<code>\@gls@numbersdef:</code> added hook	
<code>\@no@makeglossaries:</code> new	204	to set toc title	41
<code>\@print@glossary:</code> new	220	<code>\@gls@symbolsdef:</code> added hook	
<code>\@print@noidx@glossary:</code> new	232	to set toc title	41
<code>\@printgloss@setsort:</code> new	218	<code>\@glsdisp:</code> added <code>\glsifhyper</code>	146
<code>\@printglossary:</code> new	218	<code>\@glspl@:</code> added <code>\glsifhyper</code>	143
<code>almtree:</code> moved hangindent and		acronym: added hook to set toc	
parindent assignments outside		title	20
level test	338	acronyms: added hook to set toc	
<code>\compatibleglossentry:</code> changed		title	20
<code>\newcommand</code> to <code>\def</code> as is		<code>\glsdefmain:</code> added hook to set	
may or may not be defined	351	toc title	19
<code>\compatiblesubglossentry:</code>		4.07 – 2014-04-04	
changed <code>\newcommand</code> to <code>\def</code>		<code>\@glossarysection:</code> added	
as is may or may not be		optional argument when using	
defined	351	unstarred version	56
<code>\glo@grabfirst:</code> new	237	<code>\@gls@noidx@do:</code> added <code>\global</code>	
		in case it's used in a	
		tabular-like style	238

<code>\Acrfullplfmt</code> : fixed no case	
change bug	258
<code>\glsletentryfield</code> : new	170
4.08 – 2014-07-30	
General: added <code>entrycounter</code>	
option to <code>printgloss</code> family	242
added <code>nopostdot</code> option to	
<code>printgloss</code> family	242
added <code>subentrycounter</code> option	
to <code>printgloss</code> family	242
explicitly initialise hyper key	124
moved <code>\glsifhyper</code>	162–169
removed <code>\@sACRlongpl</code>	169
removed <code>\@sAcrlongpl</code>	168
removed <code>\@sacrlongpl</code>	167
removed <code>\@sACRlong</code>	167
removed <code>\@sAcrlong</code>	166
removed <code>\@sacrlong</code>	165
removed <code>\@sACRshortpl</code>	165
removed <code>\@sAcrshortpl</code>	164
removed <code>\@sacrshortpl</code>	163
removed <code>\@sACRshort</code>	163
removed <code>\@sAcrshort</code>	162
removed <code>\@sacrshort</code>	162
removed <code>\@gls@link</code>	126
removed <code>\@sGLSdescplural</code>	153
removed <code>\@sGLsdescplural</code>	153
removed <code>\@sGLSdesc</code>	152
removed <code>\@sGLsdesc</code>	152
removed <code>\@sGLSdesc</code>	152
removed <code>\@sGLSdisp</code>	145
removed <code>\@sGLSfirstplural</code>	150
removed <code>\@sGLsfirstplural</code>	150
removed <code>\@sGLsfirstplural</code>	150
removed <code>\@sGLSfirst</code>	148
removed <code>\@sGLsfirst</code>	148
removed <code>\@sGLsfirst</code>	148
removed <code>\@sGLSname</code>	151
removed <code>\@sGLsname</code>	151
removed <code>\@sGLsname</code>	151
removed <code>\@sGLSplural</code>	149
removed <code>\@sGLsplural</code>	149
removed <code>\@sGLsplural</code>	149
removed <code>\@sGLSpl</code>	145
removed <code>\@sGLspl</code>	144
removed <code>\@sGLspl</code>	143
removed <code>\@sGLSsymbolplural</code>	155
removed <code>\@sGLsymbolplural</code>	155
removed <code>\@sGLsymbolplural</code>	155
removed <code>\@sGLSsymbol</code>	154
removed <code>\@sGLsymbol</code>	154
removed <code>\@sGLsymbol</code>	154
removed <code>\@sGLStext</code>	147
removed <code>\@sGLstext</code>	147
removed <code>\@sGLstext</code>	147
removed <code>\@sGLSuseriii</code>	158
removed <code>\@sGLsuseriii</code>	158
removed <code>\@sGLsuseriii</code>	158
removed <code>\@sGLSuserii</code>	157
removed <code>\@sGLsuserii</code>	157
removed <code>\@sGLsuserii</code>	157
removed <code>\@sGLSuseriv</code>	159
removed <code>\@sGLsuseriv</code>	159
removed <code>\@sGLsuseriv</code>	159
removed <code>\@sGLSuseri</code>	156
removed <code>\@sGLsuseri</code>	156
removed <code>\@sGLsuseri</code>	156
removed <code>\@sGLSuservi</code>	161
removed <code>\@sGLsuservi</code>	161
removed <code>\@sGLsuservi</code>	161
removed <code>\@sGLSuserv</code>	160
removed <code>\@sGLsuserv</code>	160
removed <code>\@sGLsuserv</code>	160
removed <code>\@sGLS</code>	142
removed <code>\@sGLs</code>	142
removed <code>\@sGLs</code>	141
removed <code>\@thirdofthree</code>	
(defined in kernel)	140
removed <code>sPGLS</code>	349
removed <code>sPGLs</code>	347
removed <code>spgls</code>	346
removed <code>sPGLSpl</code>	349
removed <code>sPGLspl</code>	348
removed <code>spglspl</code>	346
<code>\@ACRlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	370
<code>\@ACRshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	369
<code>\@Acrlong</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	370
<code>\@Acrshort</code> : added	
<code>\do@gls@link@checkfirsthyper</code>	369
<code>\@GLS@</code> : moved <code>\glsifhyper</code>	143
moved check for first use to	
<code>\@gls@link</code>	143

\@GLSp1: moved \glsifhyper ..	145	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	259
\@gls@link	145	switched to using	
\@Gls0: moved \glsifhyper ...	142	\@gls@hyp@opt	259
moved check for first use to		\Acrfullpl: removed	
\@gls@link	142	\s@Acrfullpl	259
\@Glspl0: moved \glsifhyper .	144	switched to using	
moved check for first use to		\@gls@hyp@opt	259
\@gls@link	144	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	259
\do@gls@link@checkfirsthyper		switched to using	
.....	370	\@gls@hyp@opt	259
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	167
.....	369	\Acrlong: switched to using	
\@closegls: new	195	\@gls@hyp@opt	166
\@gls0: moved \glsifhyper ...	141	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	165
\@gls@link	141	\ACRlongpl: switched to using	
\@gls@automake: new	195	\@gls@hyp@opt	169
\@gls@doautomake: new	38	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	168
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	167
.....	146	\ACRshort: switched to using	
\@gls@forbidtextext: new	75	\@gls@hyp@opt	163
\@gls@hyp@opt: new	125	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	162
redundancy	127	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	161
\glstype	127	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	165
new	126	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	146	\@gls@hyp@opt	164
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	146	\@gls@hyp@opt	163
\@glspl0: moved \glsifhyper .	143	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	143	instead of symbol	310
\@ignored@glossaries: new ..	78	automake: new	37
\ACRfull: removed \s@ACRfull	259	\forallacronyms: new	67
switched to using		\GLS: switched to using	
\@gls@hyp@opt	258	\@gls@hyp@opt	142
\Acrfull: removed \s@Acrfull	258	\Gls: switched to using	
switched to using		\@gls@hyp@opt	141
\@gls@hyp@opt	258	\gls: switched to using	
\acrfull: removed \s@acrfull	258	\@gls@hyp@opt	141
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	258	check for ignored glossary .	100
		\gls@istfilebase: new	52

\glsaddkey: removed		\glsname: switched to using	
\@sGLS@user@{key}	94	\@gls@hyp@opt	150
removed \@sGls@user@{key}	93	\GLSpl: switched to using	
removed \@sgls@user@{key}	93	\@gls@hyp@opt	145
switched to using		\Glspl: switched to using	
\@gls@hyp@opt	93, 94	\@gls@hyp@opt	144
\GLSdesc: switched to using		\glspl: switched to using	
\@gls@hyp@opt	152	\@gls@hyp@opt	143
\Glsdesc: switched to using		\GLSplural: switched to using	
\@gls@hyp@opt	152	\@gls@hyp@opt	149
\glsdesc: switched to using		\Glsplural: switched to using	
\@gls@hyp@opt	151	\@gls@hyp@opt	149
\GLSdescplural: switched to		\glsplural: switched to using	
using \@gls@hyp@opt	153	\@gls@hyp@opt	148
\Glsdescplural: switched to		\glspace: new	257
using \@gls@hyp@opt	153	\GLSsymbol: switched to using	
\glsdescplural: switched to		\@gls@hyp@opt	154
using \@gls@hyp@opt	152	\Glsymbol: switched to using	
\glsdisablehyper: added		\@gls@hyp@opt	154
\KV@glslink@hyperfalse to		\glsymbol: switched to using	
definition	140	\@gls@hyp@opt	153
\glsdisp: switched to using		\GLSsymbolplural: switched to	
\@gls@hyp@opt	145	using \@gls@hyp@opt	155
\glsdohyperlink: new	139	\Glsymbolplural: switched to	
\glsdohypertarget: new	138	using \@gls@hyp@opt	155
\glsenablehyper: added		\glsymbolplural: switched to	
\KV@glslink@hypertrue to		using \@gls@hyp@opt	154
definition	140	\GLStext: switched to using	
\GLSfirst: switched to using		\@gls@hyp@opt	147
\@gls@hyp@opt	148	\Glstext: switched to using	
\Glsfirst: switched to using		\@gls@hyp@opt	147
\@gls@hyp@opt	148	\glstext: switched to using	
\glsfirst: switched to using		\@gls@hyp@opt	147
\@gls@hyp@opt	148	\glstreenamefmt: new	332
\GLSfirstplural: switched to		\GLSuseri: switched to using	
using \@gls@hyp@opt	150	\@gls@hyp@opt	156
\Glsfirstplural: switched to		\Glsuseri: switched to using	
using \@gls@hyp@opt	150	\@gls@hyp@opt	156
\glsfirstplural: switched to		\glsuseri: switched to using	
using \@gls@hyp@opt	149	\@gls@hyp@opt	155
\glsifhyper: deprecated	125	\GLSuserii: switched to using	
\glslink: switched to using		\@gls@hyp@opt	157
\@gls@hyp@opt	126	\Glsuserii: switched to using	
\glslinkcheckfirsthyperhook:		\@gls@hyp@opt	157
new	127	\glsuserii: switched to using	
\glslinkvar: new	125	\@gls@hyp@opt	156
\GLSname: switched to using		\GLSuseriii: switched to using	
\@gls@hyp@opt	151	\@gls@hyp@opt	158
\Glsname: switched to using		\Glsuseriii: switched to using	
\@gls@hyp@opt	151	\@gls@hyp@opt	158

<code>\glsuseriii</code> : switched to using <code>\@gls@hyp@opt</code>	157	<code>\@gls@glossary</code> : Renamed <code>\@glossary</code> to <code>\@gls@glossary</code>	206
<code>\GLSuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	159	<code>alttree</code> : moved space after symbol	339, 340
<code>\Glsuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	159	<code>\glspercentchar</code> : new	183
<code>\glsuseriv</code> : switched to using <code>\@gls@hyp@opt</code>	158	<code>\glstildechar</code> : new	183
<code>\GLSuserv</code> : switched to using <code>\@gls@hyp@opt</code>	160	4.11 – 2014-09-01 <code>\@do@esc@wrglossary</code> : added hook	212
<code>\Glsuserv</code> : switched to using <code>\@gls@hyp@opt</code>	160	<code>\gls@wrglossary</code> : renamed from <code>\@wrglossary</code> to <code>\gls@wrglossary</code>	206
<code>\glsuserv</code> : switched to using <code>\@gls@hyp@opt</code>	159	<code>\glsaddprotectedpagefmt</code> : new	208
<code>\GLSuservi</code> : switched to using <code>\@gls@hyp@opt</code>	161	<code>\glsbackslash</code> : new	183
<code>\Glsuservi</code> : switched to using <code>\@gls@hyp@opt</code>	161	<code>sanitize</code> : none option	29
<code>\glsuservi</code> : switched to using <code>\@gls@hyp@opt</code>	160	4.12 – 2014-11-22 <code>\@gls@addpredefinedattributes</code> : Added <code>glsignore</code> attribute . .	61
<code>\ifignoredglossary</code> : new	78	<code>\@gls@adjustmode</code> : new	181
<code>\newglossary</code> : added starred version	76	<code>\@gls@notranslatorhook</code> : removed	30
<code>\newignoredglossary</code> : new . . .	78	<code>\@gls@toc</code> : added <code>\protect</code> to <code>\numberline</code>	57
<code>\ns@newglossary</code> : added <code>\@glotype@{name}@log</code>	76	<code>\@gls@usetranslator</code> : new . . .	30
new	76	<code>\glsacrpluralsuffix</code> : new . . .	48
<code>\p@gls@hyp@opt</code> : new	125	<code>\glsadd</code> : added check for vertical mode	181
<code>\PGLS</code> : changed to use <code>\@gls@hyp@opt</code>	349	<code>\glsaddallunused</code> : replaced <code>@gobble</code> with <code>glsignore</code> . . .	182
<code>\Pgls</code> : changed to use <code>\@gls@hyp@opt</code>	347	<code>\glsifusedtranslatordict</code> : new	30
<code>\ppls</code> : changed to use <code>\@gls@hyp@opt</code>	346	<code>\glsignore</code> : new	182
<code>\PGLSpl</code> : changed to use <code>\@gls@hyp@opt</code>	349	<code>\glsupacrpluralsuffix</code> : new .	48
<code>\Pglspl</code> : changed to use <code>\@gls@hyp@opt</code>	348	<code>\ProvidesGlossariesLang</code> : new	49
<code>\pplspl</code> : changed to use <code>\@gls@hyp@opt</code>	346	<code>\RequireGlossariesLang</code> : new .	49
<code>\s@gls@hyp@opt</code> : new	125	4.13 – 2015-02-03 <code>\indexspace</code> : new . . .	291, 312, 332
<code>\s@newglossary</code> : new	76	4.14 – 2015-02-28 <code>\@@glslocalreset</code> : new	110
4.09 – 2014-08-12 <code>\glsaddkey</code> : fixed bug in user commands	93	<code>\@@glslocalunset</code> : new	110
4.10 – 2014-08-27 <code>\@Gls@acrentryname</code> : new	171	<code>\@@glsreset</code> : new	111
<code>\@Gls@entryname</code> : new	171	<code>\@@glsunset</code> : new	110
		<code>\@@newglossaryentry@defcounters</code> : new	112
		<code>\@cGls</code> : new	115
		<code>\@cGls@</code> : new	115
		<code>\@cGlspl@</code> : new	116
		<code>\@cgls</code> : new	115
		<code>\@cgls@</code> : new	115
		<code>\@cglspl</code> : new	116

\cglspl@: new	116	\@GLSpl: added	
\@gls@entry@count: new	115	\glspostlinkhook	145
\@gls@increment@currcount:		\@Gls@: added	
new	114	\glspostlinkhook	142
\@gls@local@increment@currcount:		\@GLspl@: added	
new	114	\glspostlinkhook	144
\@gls@write@entrycounts: new	114	\@acrlong: added	
\@glslocalreset: new	110	\glspostlinkhook	370
\@glslocalunset: new	110	\@acrshort: added	
\@glsreset: new	110	\glspostlinkhook	369
\@glsunset: new	110	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	141
new	104	\@gls@@link: added	
\cGls: new	115	\glspostlinkhook	126
\cgl: new	115	\@gls@field@link: added	
\cGlsformat: new	116	\glspostlinkhook	146
\cglformat: new	115	\@gls@link: moved definition of	
\cGLspl: new	116	\glsifhyperon outside of this	
\cglspl: new	116	macro	128
\cGlsplformat: new	117	\@glsdisp: added	
\cglsplformat: new	116	\glspostlinkhook	146
\gls@defdocnewglossaryentry:		\@glspl@: added	
new	87	\glspostlinkhook	144
\glsenableentrycount: new	112	\glsacspace: new	264
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	109	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	181
\@glslocalunset	110	\glsfielddef: new	95
\glsreset: switched to		\glsfieldedef: new	95
\@glsreset	109	\glsfieldfetch: new	96
\glsunset: switched to		\glsfieldgdef: new	95
\@glsunset	109	\glsfieldxdef: new	94
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	127
\@glo@type with \glstype	168	\glslinkpostsetkeys: new	127
4.16 – 2015-06-18		\glspostlinkhook: new	126
\glsaddstoragekey: new	91	\glswriteentry: new	207
4.16 – 2015-07-08		\ifglsfieldcseq: new	97
General: added		\ifglsfielddefeq: new	97
\glspostlinkhook	162–169	\ifglsfieldeq: new	96
\@ACRlong: added		long-sp-short: new	264
\glspostlinkhook	371	long-sp-short-desc: new	266
\@ACRshort: added		\showglofield: new	283
\glspostlinkhook	370	4.18 – 2015-09-09	
\@Acrlong: added		General: split mfirstuc into	
\glspostlinkhook	370	separate bundle	4
\@Acrshort: added		4.19 – 2015-10-31	
\glspostlinkhook	369	\glstreenamebox: new	338
\@GLS@: added			
\glspostlinkhook	143		

4.19 – 2015-11-22			
\@gls@link@nocheckfirsthyper:		\glsentrytitlecase: new	175
new	146	\glsfindwidesttoplevelname:	
\@gls@preglossaryhook: new . .	218	new	337
\@printglossary: added		\glslistgroupheaderfmt: new . .	291
\@gls@preglossaryhook . .	220	\glslistnavigationitem: new . .	292
\do@glsdisablehyperinlist:		\glstreegroupheaderfmt: new . .	332
new	127	\glstreenavigationfmt: new . .	332
\doifglossarynoexistsordo:		\ifglswrallowprimitivemods:	
new	70	new	210
\gls@gobbleopt: new	75	list: fixed missing space before	
\glsdoifexistsordo: new	70	description	293
4.20 – 2015-11-30		long: fixed typo in	
\@gls@link: added		\glossentrydesc	296
\@gls@setdefault@glslink@opts		mcolalttreesspannav: new	317
.	128	mcolindexspannav: new	313
added \glsdonohyperlink		mcoltreenonamespannav: new . .	316
when hyperlink is suppressed	128	mcoltreesspannav: new	315
\@gls@setdefault@glslink@opts:		super4col: fixed bug in	
new	127	\glossentry	322
\gls@checkseeallowed@preambleonly:		4.23 – 2016-04-30	
new	81	altlongragged4col: check for	
\glsdonohyperlink: new	139	nogroupskip changed	311
4.21 – 2016-01-24		altsuperragged4col: check for	
General: changed checkfirsthyper		nogroupskip changed	330
assignment	162–169	\glscurrentfieldvalue: new . .	74
\@printglossary: warn if no		\ifglshasfield: added	
style has been set	218	\glscurrentfieldvalue	73, 74
altlong4col-booktabs: new . .	304	long: check for nogroupskip	
altlongragged4col-booktabs:		changed	296
new	305	long-booktabs: check for	
\glsLTpenaltycheck: new	305	nogroupskip changed	302
\glspatchLoutput: new	306	long3col: check for nogroupskip	
\glspenaltygroupskip: new . .	305	changed	298
long-booktabs: new	302	long3col-booktabs: check for	
long3col-booktabs: new	303	nogroupskip changed	303
long4col-booktabs: new	303	long4col: check for nogroupskip	
longragged-booktabs: new . . .	304	changed	299
longragged3col-booktabs: new	304	long4col-booktabs: check for	
\setglossarystyle: set default		nogroupskip changed	304
style if not already set	250	longragged: check for	
4.22 – 2016-04-19		nogroupskip changed	308
\@@do@esc@wrglossary: added		longragged3col: check for	
check for \@arabic	211	nogroupskip changed	309
added test to allow temporary		super: check for nogroupskip	
primitive modifications and		changed	319
added arabic case	211	super3col: check for nogroupskip	
\gls@arabicpage: new	208	changed	321
\gls@protected@pagefmts:		super4col: check for nogroupskip	
added arabic to list	207	changed	323

superragged: check for nogroupskip changed	326	outside of theglossary environment	236
superragged3col: check for nogroupskip changed	328	\glsnavhyperlinkname: new	285
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	193	\@glo@autosee: new	104
\@gls@glossary: added check for debug mode	206	\@glo@autoseehook: new	104
\@gls@see@noindex: new	9	\@glo@check@sortallowed: new	16
debug: new	7	\@gls@noidx@do: letter group assignment made global	238
\glsnomakeindexwarning: new	58	\@gls@setupsort@def: added check for register	17
\GlsSetQuote: new	190	\@gls@setupsort@none: new	18
\GlsSetWriteIstHook: new	190	\@xdycrossrefhook: new	63
seenindex: new	9	\@xdylocationclassorder: bug fix: changed \edef to \def	64
4.25 – 2016-06-09		\glosortentrieswarning: new	23
General: changed		\gls@set@xr@key: new	81
\DeclareRobustCommand to \newrobustcmd and changed		\gls@xr@key: new	81
\ifundefined to \ifcsundef	363	\GlsAddXdyLocation: bug fix: changed #1 to #2	64
\@gls@enablesavenonumberlist: new	82	\glsnoidxstripaccents: added \a	27
\@gls@initnnonumberlist: new	82	added \TH, \dh and \DH	27
\@gls@savenonumberlist: new	82	4.31 – 2017-08-10	
4.26 – 2016-10-12		nolist: added check for “list” style	13
\@glossary@default@style: added check for classicthesis	11	4.31 – 2017-09-10	
\glstreechildpredesc: new	333	style: changed \renewcommand to \def	11
\glstreeitem: new	332	4.32 – 2017-08-24	
\glstreepredesc: new	332	\@glsnavhypertarget: new	287
\glstreesubitem: new	332	\@glsshowtarget: new	8
\glstreesubsubitem: new	332	\glsshowtarget: new	8
mcolindex: replaced \@idxitem with \glstreeitem	313	4.33 – 2017-09-20	
mcolindexspannav: replaced \@idxitem with \glstreeitem	314	\@do@esc@wrglossary: added \gls@the and \gls@number	211
4.28 – 2017-01-07		renamed from \@do@esc@wrglossary	210
\glspatchtabularx: new	109	\@do@noesc@wrglossary: new	209
4.29 – 2017-01-19		\@do@wrglossary: changed to check for esclocations	209
General: added check for \@glsxtr@doaccsupp	350	\@gls@missinglang@warn: new	23
\@gls@noidx@do: current letter group assignment made global	238	esclocations: new	12
\@gls@old@print@noidx@glossary: moved definition of \@gls@currentlettergroup		\GlsSetXdyFirstLetterAfterDigits: added starred version	183
		\GlsSetXdyNumberGroupOrder: new	184

4.34 – 2017-11-03		
<code>\glsshowtarget</code> : modified to check for math mode and inner	8	
<code>mcolaltreespannav</code> : removed spurious space	318	
4.35 – 2017-11-14		
<code>\glsadd</code> : added <code>\@gls@setsort</code> (in case of <code>sort=use</code>)	181	
4.36 – 2018-03-07		
<code>\@gls@glossary</code> : removed <code>\index</code>	206	
4.37 – 2018-04-07		
<code>\gls@begindocdefs</code> : new	89	
4.38 – 2018-05-10		
<code>\@gls@define@glossaryentrycounter</code> : added check for existence of glossaryentry counter	14	
new	14	
<code>\@gls@define@glossarysubentrycounter</code> : new	15	
prepended <code>\currentglossary.</code> to <code>\theHglossarysubentry</code> and removed spurious eol space	15	
<code>\glsaccsupp</code> : added braces around actual text argument	359	
<code>\glsentrycounterlabel</code> : bug fix: move conditional inside command	245	
<code>\GlsEntryCounterLabelPrefix</code> : new	242	
<code>\glsentryitem</code> : bug fix: move conditional inside command	245	
<code>\glsrefentry</code> : bug fix: move conditional inside command	244	
<code>\glsresetentrycounter</code> : bug fix: move conditional inside command	244	
<code>\glsresetsubentrycounter</code> : bug fix: move conditional inside command	244	
<code>\glsstepentry</code> : bug fix: move conditional inside command	244	
<code>\glsstepsubentry</code> : bug fix: move conditional inside command	244	
<code>\glssubentrycounterlabel</code> : bug fix: move conditional inside command	245	
		<code>\glssubentryitem</code> : bug fix: move conditional inside command
		245
		<code>\showglnonameaccess</code> : bug fix: corrected field (was showing text access field)
		387
4.40 – 2018-06-01		
<code>\istfile</code> : changed <code>\def</code> to <code>\providecommand</code>	204	
<code>\makenoidxglossaries</code> : false . .	201	
4.41 – 2018-07-23		
General: changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>	241	
<code>\@gls@override@glossary</code> : new	40	
debug: changed <code>\val</code> and <code>\nr</code> to <code>\gls@debug@val</code> and <code>\gls@debug@nr</code>	7	
<code>\glossary</code> : added warning	40	
<code>\gls@original@glossary</code> : new . .	39	
<code>\gls@original@makeglossary</code> : new	39	
<code>kernelglossredefs</code> : new	40	
<code>\makeglossaries</code> : removed redefinition of <code>\makeglossary</code>	199	
<code>\makeglossary</code> : added warning	39	
<code>nonumberlist</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@nonumberlist@val</code> and <code>\gls@nonumberlist@nr</code>	82	
<code>numberedsection</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@numberedsection@val</code> and <code>\gls@numberedsection@nr</code>	10	
<code>seenoindex</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@seenoindex@val</code> and <code>\gls@seenoindex@nr</code>	9	
<code>translate</code> : changed <code>\val</code> and <code>\nr</code> to <code>\gls@translate@val</code> and <code>\gls@translate@nr</code>	30	
4.42 – 2019-01-06		
<code>\@gls@@automake@immediate</code> : new	197	
<code>\@gls@automake@immediate</code> : new	196	
<code>altlong4col-booktabs</code> : removed superfluous <code>\glspatchLToutput</code>	304	

automake: switch from boolean to choice	37	\@GLSuserv@: added accessibility support	375
\gls@automake@nr: new	37	\@GLSuserii@: added accessibility support	376
\glsfielddedef: changed from \edef to \protected@csedef	95	\@GLSdesc@: added accessibility support	373
\glsfieldxdef: changed from \edef to \protected@csxdef	94	\@GLSdescplural@: added accessibility support	373
\ifglsautomake: now defined explicitly instead of through boolean key	37	\@GLSfirst@: added accessibility support	371
noglossaryindex: new	42	\@GLSfirstplural@: added accessibility support	372
4.43 – 2019-09-28		\@GLSname@: added accessibility support	372
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01	27	\@GLSplural@: added accessibility support	372
4.44 – 2019-12-06		\@GLssymbol@: added accessibility support	373
\glsprefix@record@hook: new	345	\@GLssymbolplural@: added accessibility support	374
4.45 – 2020-02-13		\@GLStext@: added accessibility support	371
General: removed backward compatibility use of symbol key	354	\@GLSuseri@: added accessibility support	374
\@do@write@glslabels: new	36	\@GLSuseriii@: added accessibility support	375
\@glsshowtarget: new	8	\@GLSuseriv@: added accessibility support	375
\@GLSdesc@: added accessibility support	373	\@GLSuserv@: added accessibility support	375
\@GLSdescplural@: added accessibility support	373	\@GLSuserii@: added accessibility support	374
\@GLSfirst@: added accessibility support	371	\@GLSuseriii@: added accessibility support	375
\@GLSfirstplural@: added accessibility support	372	\@GLSuseriv@: added accessibility support	375
\@GLSname@: added accessibility support	372	\@GLSdesc@: added accessibility support	373
\@GLSplural@: added accessibility support	372	\@GLSdescplural@: added accessibility support	373
\@GLSsymbol@: added accessibility support	373	\@gls@access@display: deprecated	360
\@GLSsymbolplural@: added accessibility support	374	\@gls@fieldaccess@display: new	360
\@GLStext@: added accessibility support	371	\@glsdesc@: added accessibility support	373
\@GLSuseri@: added accessibility support	374	\@glsdescplural@: added accessibility support	373
\@GLSuserii@: added accessibility support	374	\@glsfirst@: added accessibility support	371
\@GLSuseriii@: added accessibility support	375	\@glsfirstplural@: added accessibility support	372
\@GLSuseriv@: added accessibility support	375	\@glsname@: added accessibility support	372

\@glsplural@: added accessibility support	372	\Glsentryprefixfirstplural: added \glsdetoklabel	344
\@glsshowaccsupp: new	9	\glsentryprefixfirstplural: added \glsdetoklabel	344
\@glsymbol@: added accessibility support	373	\Glsentryprefixplural: added \glsdetoklabel	345
\@glsymbolplural@: added accessibility support	374	\glsentryprefixplural: added \glsdetoklabel	344
\@glstext@: added accessibility support	371	\glsentrytitlecase: added existence check	175
\@glsuseri@: added accessibility support	374	\glsentryuseriaccess: new	357
\@glsuserii@: added accessibility support	374	\glsentryuseriiaccess: new	357
\@glsuseriii@: added accessibility support	375	\glsentryuseriiiaccess: new	358
\@glsuseriv@: added accessibility support	375	\glsentryuserivaccess: new	358
\@glsuserv@: added accessibility support	375	\glsentryuserviaccess: new	358
\@glsuservi@: added accessibility support	376	\glsfieldaccsupp: new	358
\@org@Gls@acrentryname: added check for \glsshortaccessdisplay	172	\glsfirstaccessdisplay: added check for existence	360
acronyms: changed \renewcommand to \def	20	\glsfirstpluralaccessdisplay: added check for existence	361
debug: showaccsupp	7	\glslongaccessdisplay: added check for existence	362
disablemakegloss: new	35	\glslongpluralaccessdisplay: added check for existence	362
\gls@accessibility: new	351	\glsnameaccessdisplay: added check for existence	360
\gls@accsupp@engine: new	351	\glspluralaccessdisplay: added check for existence	360
\glsaccessibility: new	351	\glsprefixsep: new	346
\glsdefaultshortaccess: new	386	\glssee: switched to \newrobustcmd	215
\glsdescriptionaccessdisplay: added check for existence	361	\glsseeformat: switched to \newrobustcmd	215
\glsdescriptionpluralaccessdisplay: added check for existence	361	\glsseeitem: switched to \newrobustcmd	216
\glsentrydescpluralaccess: corrected field reference	357	\glsseelist: switched to \newrobustcmd	215
\glsentryfirstpluralaccess: switched to using \@gls@entry@field	356	\glsshortaccessdisplay: added check for existence	361
\glsentryparent: new	176	\glsshortaccsupp: new	359
\Glsentryprefix: added \glsdetoklabel	344	\glsshortplaccsupp: new	359
\glsentryprefix: added \glsdetoklabel	344	\glsshortpluralaccessdisplay: added check for existence	362
\Glsentryprefixfirst: added \glsdetoklabel	344	\glsshowaccsupp: new	9
\glsentryprefixfirst: added \glsdetoklabel	344	\glsshowtargetfont: new	8
		\glsshowtargetouter: new	8
		\glsshowtargetsymbols: new	8
		\glsymbolaccessdisplay: added check for existence	361

<code>\glssymbolpluralaccessdisplay:</code>	<code>\ifglossaryexists:</code> added
added check for existence 361	starred form 68
<code>\glstextaccessdisplay:</code> added	<code>\s@ifglossaryexists:</code> new 68
check for existence 360	<code>\setglossarypreamble:</code> switched
<code>\glsuseriaccessdisplay:</code> new . 362	to starred form of
<code>\glsuseriiaccessdisplay:</code> new 362	<code>\ifglossaryexists</code> 54
<code>\glsuseriiiaccessdisplay:</code>	4.47 – 2021-09-20
new 362	General: changed <code>\x</code> to <code>\@gls@x</code> 253
<code>\glsuserivaccessdisplay:</code> new 362	<code>\@do@esc@wrglossary:</code> changed
<code>\glsuservaccessdisplay:</code> new . 363	<code>\edef</code> to <code>\protected@edef</code> 212
<code>\glsuserviiaccessdisplay:</code> new 363	<code>\@do@noesc@wrglossary:</code>
<code>\ifglshaschildren:</code> made robust 70	changed <code>\edef</code> to
<code>\ifglshasfield:</code> made robust . 72	<code>\protected@edef</code> 210
<code>\ifglshaslong:</code> made robust . . 72	<code>\@do@write@glslabels:</code> fully
<code>\ifglshasprefix:</code> added	expand list 36
<code>\glsdetoklabel</code> 345	<code>\@gls@@default@acr@descplural:</code>
<code>\ifglshasprefixfirst:</code> added	new 275
<code>\glsdetoklabel</code> 345	<code>\@gls@@default@acr@firstplural:</code>
<code>\ifglshasprefixfirstplural:</code>	new 275
added <code>\glsdetoklabel</code> . . . 345	<code>\@gls@@default@acr@plural:</code>
<code>\ifglshasprefixplural:</code> added	new 275
<code>\glsdetoklabel</code> 345	<code>\@gls@noexpand@assign@default@plurals:</code>
<code>\ifglshasshort:</code> made robust . 72	new 275
<code>\ifglshassymbol:</code> made robust 71	<code>\glo@do@sortentries:</code> changed
<code>\makeglossaries:</code> let	<code>\edef</code> to <code>\protected@edef</code> 228
<code>\@makeglossary</code> to <code>\@gobble</code>	<code>\glo@sortentries:</code> changed
instead of <code>\relax</code> 199	<code>\edef</code> to <code>\protected@edef</code> 227
<code>restoremakegloss:</code> new 35	<code>\glo@storeentry:</code> changed
<code>user1access:</code> new 353	<code>\edef</code> to <code>\protected@edef</code> 105
<code>user2access:</code> new 353	<code>\@gls@checkesc:</code> new 136
<code>user3access:</code> new 353	<code>\@gls@deprecated@acr@warning:</code>
<code>user4access:</code> new 353	new 277
<code>user5access:</code> new 353	<code>\@gls@link:</code> changed <code>\edef</code> to
<code>user6access:</code> new 353	<code>\protected@edef</code> 127
<code>writeglslabels:</code> new 36	<code>\@gls@setupsort@use:</code> changed
<code>\xglfieldaccsupp:</code> new 359	<code>\edef</code> to <code>\protected@edef</code> . 18
4.46 – 2020-03-19	<code>\@gls@writedef:</code> added a
<code>\@ifglossaryexists:</code> new 68	warning if entry has already
<code>\@printgloss@checkexists:</code>	been defined 89
new 217	<code>\@makeglossaries@warn@noprintglossary:</code>
<code>\@printgloss@checkexists@allowignored:</code>	new 200
new 217	<code>\@write@glslabelsname:</code> new . 36
<code>\@printgloss@checkexists@noignored:</code>	<code>\do@write@glslabelsname:</code> new 36
new 217	<code>\forglsentries:</code> changed <code>\edef</code>
<code>\@printglossary:</code> replaced	to <code>\protected@edef</code> 67
<code>\ifglossaryexists</code> with	<code>\gls@defdocnewglossaryentry:</code>
<code>\@printgloss@checkexists</code> 219	removed see warning 87
<code>\doifglossarynoexistsordo:</code>	<code>\gls@defglossaryentry:</code> changed
switched to starred form of	<code>\edef</code> to
<code>\ifglossaryexists</code> 70	<code>\protected@edef</code> 99, 100

<code>\gls@docdef@addlabel</code> : new ..	87	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\gls@docdef@labellist</code> : new ..	87	added deprecated warnings	276
<code>\gls@glsdefs@addlabel</code> : new ..	87	<code>\SetDUASStyle</code> : added deprecated	
<code>\gls@glsdefs@labellist</code> : new ..	87	warnings	277
<code>\gls@save@numberlist</code> : changed		<code>\SetFootnoteAcronymStyle</code> :	
<code>\edef</code> to <code>\protected@edef</code>	216	added deprecated warnings	277
<code>\glsadd</code> : changed <code>\edef</code> to		<code>\SetSmallAcronymStyle</code> : added	
<code>\protected@edef</code>	181	deprecated warnings	277
<code>\glsaddall</code> : changed <code>\edef</code> to		<code>writeglslabelnames</code> : new	37
<code>\protected@edef</code>	182	4.48 – 2021-10-19	
<code>\glsdefs@newdocentry</code> : new ..	89	<code>\glscapitalisewords</code> : new	175
<code>\glsdisplaynumberlist</code> : changed		<code>\glslistexpandedname</code> : new ..	292
<code>\edef</code> to <code>\protected@edef</code>	180	<code>\glslistinit</code> : new	292
<code>\glsfielddef</code> : changed <code>\edef</code> to		<code>\glsunexpandedfieldvalue</code> : new	96
<code>\protected@edef</code>	95	4.49 – 2021-11-01	
<code>\glsfielddedef</code> : changed <code>\edef</code> to		<code>\glslistinit</code> : added	
<code>\protected@edef</code>	95	<code>\glsnavigation</code>	292
<code>\glsfieldfetch</code> : changed <code>\edef</code>		4.50 – 2022-10-14	
to <code>\protected@edef</code>	96	General: added initialisation ..	255
<code>\glsfieldgdef</code> : changed <code>\edef</code> to		allow markup to be stripped	
<code>\protected@edef</code>	95	from location	253
<code>\glsfieldxdef</code> : changed <code>\edef</code> to		corrected description mapping	354
<code>\protected@edef</code>	94	don't load <code>glossary-super</code> if <code>xtab</code>	
<code>\glsIfListOfAcronyms</code> : changed		has been loaded	12
<code>\edef</code> to <code>\protected@edef</code>	21	No longer requires <code>textcase</code>	
<code>\glsmoveentry</code> : changed <code>\edef</code> to		with <code>mfirstuc 2.08+</code>	5
<code>\protected@edef</code>	105	removed automatic loading of	
<code>\glsnavhyperlink</code> : changed		<code>glossaries-compatible-307</code> ..	43
<code>\edef</code> to <code>\protected@edef</code>	285	switched to using	
<code>\GlsSetQuote</code> : added check for		<code>\glsuppercase</code>	
backslash	192 147–161, 163, 165, 167, 169	
<code>\glsstepsubentry</code> : changed		<code>\@@do@esc@wrglossary</code> : moved	
<code>\edef</code> to <code>\protected@edef</code>	244	<code>\romannumeral</code> inside allow	
<code>\ifglsfieldcseq</code> : changed <code>\edef</code>		primitive mods	212
to <code>\protected@edef</code>	97	<code>\@@gls@@default@acr@descplural</code> :	
<code>\ifglshaschildren</code> : changed		removed	275
<code>\edef</code> to <code>\protected@edef</code>	71	<code>\@@gls@@default@acr@firstplural</code> :	
<code>\ifignoredglossary</code> : changed		removed	275
<code>\edef</code> to <code>\protected@edef</code>	78	<code>\@@gls@@default@acr@plural</code> :	
<code>\makenoidxglossaries</code> : changed		removed	275
<code>\edef</code> to <code>\protected@edef</code>	201	<code>\@@gls@default@entryfmt</code> :	
<code>\new@glossaryentry</code> : allow		removed	363
overwrite if defined in		<code>\@@gls@navhypertarget</code> : new ..	287
<code>.glsdefs</code>	88	<code>\@@gls@noexpand@assign@default@plurals</code> :	
<code>\newignoredglossary</code> : changed		removed	275
<code>\edef</code> to <code>\protected@edef</code>	78	<code>\@Gls@acronymname</code> : removed	
<code>\ns@newglossary</code> : changed <code>\edef</code>		<code>mfirstuc</code> internal commands	171
to <code>\protected@edef</code>	76	<code>\@Glsdisp</code> : new	146
<code>\SetDescriptionDUAAcronymStyle</code> :		<code>\@Glsentryfield</code> : new	170
added deprecated warning	276		

\@domakeglossaries: added		\CustomNewAcronymDef:	
check for \nofiles	35	removed	278
\@gls@automake: added check for		\DefaultNewAcronymDef:	
void codepage	195	removed	275, 387
\@gls@automake@immediate:		\defglsdisplay: removed	123
added check for void		\defglsdisplayfirst: removed	123
codepage	196	\DescriptionDUANewAcronymDef:	
\@gls@automake@types: new	197	removed	276
\@gls@defaulttype: new	74	\DescriptionFootnoteNewAcronymDef:	
\@gls@do@automake@err: new	37	removed	276, 387
\@gls@do@glsunset: new	124	\DescriptionNewAcronymDef:	
\@gls@entry@field: switched to		removed	276, 387
\csuse	169	dua: switched to using	
\@gls@escbsdq: check		\glsuppercase	269, 270
esclocations setting	131	dua-desc: removed \acronymfont	
\@gls@getcounterprefix:		from around long form	270
changed \edef to \def	213	\DUANewAcronymDef: removed	277
\@gls@getgrouptitle: switched		esclocations: default now false	12
to L ^A T _E X3 syntax	249	\findrootlanguage: removed	65
\@gls@glslink@hyper@update@hook:		footnote: switched to using	
new	124	\glsuppercase	271
\@gls@hypergroup: updated to		\FootnoteNewAcronymDef:	
use etoolbox	288	removed	276, 387
\@gls@noist@toolate: new	194	\glossary: removed	40
\@gls@removedoption: new	28	\glossaryentryfield: removed	247
\@gls@requested@languages:		\glossarymark: switched to using	
new	32	\glsuppercase	56
\@gls@restore@glslocal: new	124	\glossarystyle: removed	251
\@gls@sanitizedesc: removed	24	\glossarysubentryfield:	
\@gls@save@glslocal: new	124	removed	248
\@gls@setupsort@clear: new	18	\gls@automake@makegloss: new	37
\@gls@defaultplural: Removed	84	\gls@automake@option: new	37
\@org@Gls@acentryname: saved		\gls@codepage: set default	
original definition of		codepage to utf8 if	
\@Gls@acentryname	172	\inputencodingname not	
\@xp@gls@getcounterprefix:		defined and switched to	
new	213	\ifdefvoid	34
\@xp@gls@hypernumber: new	252	\gls@delimN: renamed \@delimN	
automake: added makegloss and		to \gls@delimN	253
lite options	37	\gls@delimR: renamed \@delimR	
automakegloss: new	38	to \gls@delimR	252
automakeglosslite: new	38	\gls@disablepagerefexpansion:	
compatible-2.07: deprecated	39	added check for esclocations	207
compatible-3.07: deprecated	39	\gls@noidx@nonumberlist: new	82
\compatibleglossentry:		\gls@noidx@glossary: added	
removed	246, 351	\gls@wrglossdisablelocationcmds	
\compatiblesubglossentry:			204
removed	247, 351	\gls@xdy@set@lang: new	34
\CustomAcronymFields:		\gls@addkey: allow expansion in	
removed	278	PDF bookmarks	93

switched to using		<code>\Glsentryshort</code> : allow expansion	
<code>\glssupercase</code>	94	in PDF bookmarks	178
<code>\GlsAddXdyAttribute</code> : remove		<code>\Glsentryshortpl</code> : allow	
duplicate counterless		expansion in PDF	
attribute definitions	60	bookmarks	178
<code>\Glsdisp</code> : new	146	<code>\Glsentrysymbol</code> : allow	
<code>\glssdisplay</code> : removed	123	expansion in PDF	
<code>\glssdisplayfirst</code> : removed	123	bookmarks	174
<code>\glssdoparenifnotempty</code> :		<code>\Glsentrysymbolplural</code> : allow	
replaced		expansion in PDF	
<code>\DeclareRobustCommand</code> with		bookmarks	174
<code>\newrobustcmd</code>	277	<code>\Glsentrytext</code> : allow expansion	
<code>\glssdoshowtarget</code> : new	8	in PDF bookmarks	173
<code>\glssencapwrcntent</code> : new	128	<code>\glssentrytitlecase</code> : use	
<code>\Glsentrydesc</code> : allow expansion		sentence case for PDF	
in PDF bookmarks	173	bookmark	175
<code>\Glsentrydescplural</code> : allow		<code>\Glsentryuseri</code> : allow expansion	
expansion in PDF		in PDF bookmarks	176
bookmarks	173	<code>\Glsentryuserii</code> : allow	
<code>\Glsentryfirst</code> : allow expansion		expansion in PDF	
in PDF bookmarks	174	bookmarks	176
<code>\Glsentryfirstplural</code> : allow		<code>\Glsentryuseriii</code> : allow	
expansion in PDF		expansion in PDF	
bookmarks	175	bookmarks	176
<code>\glssentryfmt</code> : obsolete		<code>\Glsentryuseriv</code> : allow	
commands replaced with		expansion in PDF	
<code>\glssgenentryfmt</code>	118	bookmarks	177
<code>\Glsentrylong</code> : allow expansion		<code>\Glsentryusersv</code> : allow expansion	
in PDF bookmarks	178	in PDF bookmarks	177
<code>\Glsentrylongpl</code> : allow		<code>\Glsentryusersvi</code> : allow	
expansion in PDF		expansion in PDF	
bookmarks	178	bookmarks	177
<code>\Glsentryname</code> : allow expansion		<code>\glssfielddef</code> : made robust	95
in PDF bookmarks	171	<code>\glssfieldedef</code> : made robust	95
<code>\Glsentryplural</code> : allow		<code>\glssfieldfetch</code> : made robust	96
expansion in PDF		<code>\glssfieldgdef</code> : made robust	95
bookmarks	174	<code>\glssfieldxdef</code> : made robust	94
<code>\Glsentryprefix</code> : allow		<code>\glssgenacfmt</code> : switched to using	
expansion in PDF		<code>\glssupercase</code>	120
bookmarks	344	<code>\glssgenentryfmt</code> : switched to	
<code>\Glsentryprefixfirst</code> : allow		using <code>\glssupercase</code>	119
expansion in PDF		<code>\glssgetgrouplabel</code> : converted to	
bookmarks	344	use \LaTeX 3	250
<code>\Glsentryprefixfirstplural</code> :		<code>\glssgetgrouptitle</code> : made	
allow expansion in PDF		robust	249
bookmarks	344	<code>\glssglossarymark</code> : switched to	
<code>\Glsentryprefixplural</code> : allow		<code>\glssupercase</code>	55
expansion in PDF		<code>\glssifhyper</code> : removed	125
bookmarks	345	<code>\glssindexingsetting</code> : new	15
		<code>\glssinlineifhaschildren</code> : new	290

<code>\Glslink</code> : new	126	<code>\SetCustomDisplayStyle</code> :	
<code>\glslocationcstoencap</code> : new	53	removed	278
<code>\glslowercase</code> : new	4	<code>\SetCustomStyle</code> : removed	278
<code>\glsmfuaddmap</code> : new	5	<code>\SetDefaultAcronymStyle</code> :	
<code>\glsmfublocker</code> : new	5	removed	275
<code>\glsmfuexcl</code> : new	5	<code>\SetDescriptionAcronymDisplayStyle</code> :	
<code>\glsnavhyperlink</code> : changed		removed	276
<code>@glo@type</code> to		<code>\SetDescriptionAcronymStyle</code> :	
<code>\currentglossary</code>	285	removed	276
<code>\glsnavhypertarget</code> : changed		<code>\SetDescriptionDUAAcronymDisplayStyle</code> :	
<code>@glo@type</code> to		removed	276
<code>\currentglossary</code>	287	<code>\SetDescriptionDUAAcronymStyle</code> :	
<code>\glsnavigation</code> : changed		removed	276
<code>@glo@type</code> to		<code>\SetDescriptionFootnoteAcronymDisplayStyle</code> :	
<code>\currentglossary</code>	288	removed	276
<code>\glsnoidxprenumberlist</code> : new	82	<code>\SetDescriptionFootnoteAcronymStyle</code> :	
<code>\glsnumberlistloop</code> : corrected		removed	276
parameter number	203	<code>\SetDUADisplayStyle</code> : removed	277
<code>\glsseeitemformat</code> : added test		<code>\SetDUASStyle</code> : removed	277
for short field	216	<code>\SetFootnoteAcronymDisplayStyle</code> :	
<code>\glsentencecase</code> : new	4	removed	276
<code>\glsshowtargetfonttext</code> : new	8	<code>\SetFootnoteAcronymStyle</code> :	
<code>\glsshowtargetinner</code> : new	8	removed	277
<code>\glstexpdfstring</code> : new	170	<code>\SetGenericNewAcronym</code> :	
<code>\glsuppercase</code> : new	4	corrected long plural suffix	261
<code>\GlsWarnAddProtectedPageFmt</code> :		switched to using	
new	208	<code>\glsuppercase</code>	261
<code>\glswrglossdisableanchorcmds</code> :		<code>\setglossentrycompatibility</code> :	
new	209	removed	247
<code>\glswrglossdisablelocationcmds</code> :		<code>\SetSmallAcronymDisplayStyle</code> :	
new	207	removed	277
<code>\glswrglosslocationtarget</code> :		<code>\SetSmallAcronymStyle</code> :	
new	209	removed	277
<code>\glswrglosslocationtextfmt</code> :		<code>\SmallNewAcronymDef</code> :	
new	209	removed	277, 387
<code>\ifglsfieldcseq</code> : made robust	97	<code>sort</code> : added clear	15
<code>\ifglsfielddefeq</code> : made robust	97	4.51 – 2022-11-02	
<code>\ifglsfieldeq</code> : made robust	96	General: switch to	
<code>\ifglsfieldvoid</code> : new	72	<code>\TrackIfKnownLanguage</code>	49
<code>\ifglsresetcurrcount</code> : new	112	<code>\do@wrglossary</code> : added	
<code>\ifglsrallowprimitivemods</code> :		<code>\gls@ifnotmeasuring</code>	207
switch default to false	210	<code>@gls@@automake@immediate</code> :	
<code>\istfile</code> : removed	204	ensure list is expanded before	
<code>kernelglossredefs</code> : corrected		iterating	197
names of choice variables	40	<code>@gls@if@measuring</code> : new	107
<code>languages</code> : new	32	<code>alttree</code> : replaced <code>\settowidth</code>	
<code>\makeglossary</code> : removed	39	with <code>\glsmeasurewidth</code>	338, 339
<code>mfirstuc</code> : new	42	<code>\gls@if@measuring</code> : new	107
<code>\SetAcronymStyle</code> : removed	278	<code>\gls@local@stepcounter</code> : new	107
		<code>\gls@org@refstepcounter</code> : new	107

<code>\gls@start@measuring</code> : new ..	108	<code>\Glsentryplural</code> : change map to exclusion	174
<code>\gls@stop@measuring</code> : new ...	108	<code>\Glsentryshort</code> : change map to exclusion	178
<code>\glsdohypertarget</code> : added measuring and group	138	<code>\Glsentryshortpl</code> : change map to exclusion	178
<code>\glsfindwidesttoplevelname</code> : replaced <code>\settowidth</code> with <code>\glsmeasurewidth</code>	337	<code>\Glsentrysymbol</code> : change map to exclusion	174
<code>\glsifmeasuring</code> : new	107	<code>\Glsentrysymbolplural</code> : change map to exclusion	174
<code>\glsmeasuredepth</code> : new	108	<code>\Glsentrytext</code> : change map to exclusion	173
<code>\glsmeasureheight</code> : new	108	<code>\Glsentryuseri</code> : change map to exclusion	176
<code>\glsmeasurewidth</code> : new	109	<code>\Glsentryuserii</code> : change map to exclusion	176
<code>\glspatchtabularx</code> : switch measuring on and off	109	<code>\Glsentryuseriii</code> : change map to exclusion	176
<code>\makeglossaries</code> : ensure list of glossaries is expanded before iterating	199	<code>\Glsentryuseriv</code> : change map to exclusion	177
4.52 – 2022-11-03		<code>\Glsentryuseriv</code> : change map to exclusion	177
<code>\gls@local@stepcounter</code> : corrected spelling of <code>\@ne</code> ..	107	<code>\Glsentryuseriv</code> : change map to exclusion	177
4.53 – 2023-09-29		<code>\Glsentryuseriv</code> : change map to exclusion	177
<code>\@gls@hypergroup</code> : rewritten in \LaTeX 3	288	<code>\Glsentryuseriv</code> : change map to exclusion	177
<code>\glsnavhypergroupdotarget</code> : new	287	<code>\Glsentryuseriv</code> : change map to exclusion	177
<code>\glsnavigationitem</code> : new	288	<code>\glslabelhypertarget</code> : new ..	138
4.54 – 2024-04-03		<code>\glslabelhypertargetdefs</code> : new	139
General: added check for <code>\l_datatool_measure_hook_tl</code>	108	<code>\glslabelhypertargetprefix</code> : new	139
<code>\gls@printglossary@begin</code> : new	220	<code>\glslabelhypertargetvalue</code> : new	139
<code>\gls@printglossary@end</code> : new ..	220	<code>\glslowercase</code> : ensure argument is expanded	4
<code>\glsdohyperlinkhook</code> : new ...	139	<code>\glsuppercase</code> : ensure argument is expanded	4
<code>\glsdohypertargethook</code> : new ..	138	4.55 – 2024-11-01	
<code>\Glsentrydesc</code> : change map to exclusion	173	<code>\@gls@run@output@dir</code> : new ..	43
<code>\Glsentrydescplural</code> : change map to exclusion	173	<code>\@gls@run@output@file</code> : new ..	43
<code>\Glsentryfirst</code> : change map to exclusion	174	<code>\@gls@run@unrestricted@shell</code> : new	43
<code>\Glsentryfirstplural</code> : change map to exclusion	175	<code>\@gls@runshell</code> : new	43
<code>\Glsentryfull</code> : change map to exclusion	179	<code>\gls@set@xr@key</code> : added check for empty value	81
<code>\Glsentrylong</code> : change map to exclusion	178	locales: new	32
<code>\Glsentrylongpl</code> : change map to exclusion	178	4.56 – 2025-03-03	
<code>\Glsentryname</code> : change map to exclusion	171	General: store original sort as well as label in actual part	222
		4.57 – 2025-03-12	
		<code>\@@gls@noidx@nosanitizesort</code> : check for datatool preprocess	

function	26	token list	232
\@print@noidx@glossary: switched to constructing		\glsnoidxprecontenthook: new	233

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

Symbols	
<code>\!</code>	132, 134, 192, 464, 713, 715, 745, 966, 968, 969, 1000, 1121, 1123, 1124, 1155
<code>\"</code>	27, 131, 133, 134, 137, 401, 462–464, 466, 651, 712–715, 717, 718, 905, 966–969, 971, 1060, 1121–1124, 1126
<code>\#</code>	136, 137, 187, 489, 717, 741, 970, 971, 995, 1125, 1126, 1151
<code>\%</code>	183, 189, 487, 491, 592, 593, 738, 742, 743, 845, 846, 993, 997, 1148, 1153
<code>\&</code>	48, 180, 412, 486, 662, 737, 917, 992, 1072, 1148
<code>\'</code>	27, 401, 651, 905, 1060
<code>\.</code>	13, 27, 394, 401, 644, 651, 898, 905, 1052, 1060
<code>\=</code>	27, 401, 651, 905, 1060
<code>\?</code>	131, 133, 192, 462, 463, 493, 712, 714, 745, 966–968, 999, 1121–1123, 1155
<code>\@</code>	89, 434, 684, 940, 1095
<code>\@@delimN</code>	524, 775, 776
<code>\@do@wrglossary</code>	201, 210, 212, 212, 499, 504, 505, 751, 756, 757, 1006, 1012, 1162, 1167, 1168
<code>\@do@esc@wrglossary</code>	209, 210, 504, 756, 1011, 1012, 1167
<code>\@do@noesc@wrglossary</code>	209, 209, 504, 756, 1011, 1167
<code>\@do@wrglossary</code>	181, 207, 209, 486, 503, 504, 738, 755, 756, 992, 1010, 1011, 1148, 1166, 1167
<code>\@do@write@glslabels</code>	36, 36, 407, 656, 911, 1066
<code>\@glo@sortentries</code>	227, 230, 231
<code>\@glo@assign@sortkey</code>	201, 243, 500, 518, 751, 770, 1007, 1025, 1162, 1181
<code>\@glo@list</code>	67, 423, 673, 928, 1083
<code>\@glo@sort</code>	26, 401, 650, 905, 1059
<code>\@glo@type</code>	218, 508, 760, 1015, 1016, 1171
<code>\@glossarysec</code>	10, 10, 56, 57, 392, 393, 415, 416, 642, 665, 666, 896, 921, 922, 1051, 1076
<code>\@glossaryseclabel</code>	10, 10, 11, 56, 57, 241, 242, 393, 416, 517, 642, 666, 769, 896, 897, 921, 1024, 1025, 1051, 1076, 1180
<code>\@glossarysecstar</code>	10, 10, 11, 56, 57, 241, 242, 393, 415, 416, 517, 642, 665, 666, 769, 896, 897, 921, 1024, 1025, 1051, 1076, 1180
<code>\@gls@default@acr@desclplural</code>	275, 790
<code>\@gls@default@acr@firstplural</code>	275, 790
<code>\@gls@default@acr@plural</code>	275, 789, 790
<code>\@gls@checkactual</code>	136, 465, 466, 716, 717, 970, 1125
<code>\@gls@checkbar</code>	135, 464, 465, 715, 716, 969, 1124
<code>\@gls@checkescactual</code>	133, 463, 714, 968, 1122, 1123
<code>\@gls@checkeschar</code>	134, 464, 714, 715, 968, 1123
<code>\@gls@checkesclevel</code>	134, 135, 464, 715, 969, 1123, 1124
<code>\@gls@checkescquote</code>	133, 193, 463, 494, 714, 746, 967, 1000, 1001, 1122, 1156
<code>\@gls@checklevel</code>	135, 136, 465, 716, 970, 1124, 1125
<code>\@gls@checkquote</code>	132, 191, 462, 492, 493, 713, 744, 967, 999, 1122, 1154, 1155
<code>\@gls@codepage</code>	195, 196, 1002, 1003, 1158, 1159
<code>\@gls@default@entryfmt</code>	363, 451, 457, 562, 701, 708, 815
<code>\@gls@expand@field</code>	

.....	24, <u>85</u> , 86, 91, 92,	\@ACRfullpl	259,
	400, 432, 433, 435, 436, 537,		260, 526, 778, 1033, 1188, 1189
	539, 541, 543, 544, 547, 548,	\@ACRlong	167,
	583–587, 649, 682, 683, 686,		257, <u>370</u> , 479, 526, 570, 731,
	687, 789, 791, 793, 795, 797,		777, 823, 985, 1032, 1140, 1187
	799–801, 836–840, 904, 938,	\@ACRlongpl	169, 258, 480, 481, 526,
	939, 942, 943, 1058, 1093, 1097		732, 778, 986, 1032, 1142, 1188
\@@gls@extramakeindexopts		\@ACRshort	163,
.....	190, 198, 492, 498, 744,		257, <u>369</u> , 477, 526, 569, 729,
	749, 750, 998, 1005, 1154, 1160		777, 822, 982, 1032, 1138, 1187
\@@gls@fixbraces		\@ACRshortpl ..	165, 258, 478, 526,
	215, <u>215</u> , 507, 758, 759, 1014, 1170		730, 778, 984, 1032, 1139, 1188
\@@gls@navhypertarget	287, <u>287</u> , 1046	\@Acrfull ..	258, 525, 777, 1032, 1188
\@@gls@noexpand@assign@default@plurals	\@Acrfullpl ..	259, 526, 778, 1033, 1188
	<u>275</u> , 789, 836	\@Acrlong	166,
\@@gls@noexpand@field			257, <u>370</u> , 479, 525, 570, 730,
.....	24, 85, 400, 432, 649,		777, 823, 984, 1032, 1140, 1187
	682, 904, 938, 1058, 1092, 1093	\@Acrlongpl ...	168, 258, 480, 526,
\@@gls@noidx@no@sanitizesort ..			732, 778, 986, 1032, 1141, 1188
.....	26, 401, 650, 904, 905, 1059	\@Acrshort	162,
\@@gls@noidx@nosanitizesort ...			<u>369</u> , 477, 569, 728, 822, 982, 1138
.....	<u>26</u> , 203, 401, 501,	\@Acrshortpl ..	164, 478, 729, 983, 1139
	650, 753, 904, 1008, 1059, 1164	\@Alpha	208, 211, 212, 503–505,
\@@gls@noidx@sanitizesort			755–757, 1010, 1012, 1166–1168
.....	<u>25</u> , 400, 650, 904, 1059	\@GLS	142, 469, 720, 974, 1129
\@@gls@nosanitizesort		\@GLS@	142, <u>142</u> ,
.....	25, <u>25</u> , 203, 400, 501,		349, 469, 609, 720, 862, 974, 1129
	650, 753, 904, 1008, 1059, 1164	\@GLSdesc	152, 473, 724, 978, 1133, 1134
\@@gls@sanitizesort		\@GLSdesc@	152,
.....	25, <u>25</u> , 203, 400, 501,		<u>373</u> , 473, 571, 724, 825, 978, 1134
	650, 753, 904, 1008, 1059, 1164	\@GLSdescplural	
\@@gls@xdycheckbackslash ..	137,		153, 473, 725, 978, 1134
	138, 466, 467, 718, 972, 1126, 1127	\@GLSdescplural@	153,
\@@gls@xdycheckquote			<u>373</u> , 473, 572, 725, 825, 978, 1134
...	137, 466, 717, 718, 971, 1126	\@GLSfirst ..	148, 471, 723, 976, 1132
\@@gls@localreset	110, <u>110</u> , 113, 447,	\@GLSfirst@	148,
	448, 698, 699, 955, 956, 1109, 1111		<u>371</u> , 472, 571, 723, 824, 976, 1132
\@@gls@localunset	110, <u>110</u> , 113, 447,	\@GLSfirstplural	
	448, 697, 699, 954, 956, 1109, 1111		150, 472, 724, 977, 1133
\@@gls@reset ...	110, <u>111</u> , 113, 447,	\@GLSfirstplural@	150,
	448, 698, 699, 955, 956, 1109, 1111		<u>372</u> , 472, 571, 724, 824, 977, 1133
\@@gls@showtarget ...	7, <u>8</u> , 9, 391,	\@GLSname	151, 473, 724, 977, 978, 1133
	392, 640, 641, 894, 895, 1049, 1050	\@GLSname@	151,
\@@gls@unset ...	110, <u>110</u> , 113, 447,		<u>372</u> , 473, 571, 724, 824, 978, 1133
	448, 698, 699, 954, 956, 1109, 1111	\@GLSpl ..	145, <u>145</u> , 470, 721, 975, 1130
\@@newglossaryentry@defcounters		\@GLSpl@	145,
...	112, <u>112</u> , 448, 698, 955, 1110		350, 470, 609, 721, 862, 975, 1130
\@this@glo@ ..	68, 423, 673, 929, 1083	\@GLSplural ..	149, 472, 723, 977, 1132
\@ACRfull ..	259, 526, 777, 1032, 1188	\@GLSplural@	149,
			<u>372</u> , 472, 571, 723, 824, 977, 1132

<code>\@GLSsymbol</code> .. 154, 474, 725, 979, 1134	<code>\@Glsdesc</code> .. 152,
<code>\@GLSsymbol@</code> .. 154,	373, 473, 571, 724, 824, 978, 1133
373, 474, 572, 725, 825, 979, 1134	<code>\@Glsdescplural</code> ..
<code>\@GLSsymbolplural</code> ..	153, 473, 725, 978, 1134
155, 474, 725, 726, 979, 1135	<code>\@Glsdescplural@</code> .. 153,
<code>\@GLSsymbolplural@</code> .. 155,	373, 473, 572, 725, 825, 978, 1134
374, 474, 572, 726, 825, 979, 1135	<code>\@Glsdisp</code> .. 146, 146, 975, 1131
<code>\@GLStext</code> 147, 471, 722, 723, 976, 1131	<code>\@Glsentryfield</code> .. 93,
<code>\@GLStext@</code> .. 147,	170, 173–178, 344, 345, 943,
371, 471, 570, 723, 823, 976, 1131	987–991, 1098, 1142, 1144–1146
<code>\@GLSuseri</code> 156, 474, 726, 979, 980, 1135	<code>\@Glsfirst</code> .. 148, 471, 723, 976, 1132
<code>\@GLSuseri@</code> .. 156,	<code>\@Glsfirst@</code> .. 148,
374, 474, 572, 726, 825, 980, 1135	371, 471, 571, 723, 824, 976, 1132
<code>\@GLSuserii</code> ..	<code>\@Glsfirstplural</code> ..
157, 475, 726, 980, 1135, 1136	150, 472, 724, 977, 1133
<code>\@GLSuserii@</code> .. 157,	<code>\@Glsfirstplural@</code> ..
374, 475, 573, 726, 826, 980, 1136	150, 372, 472, 724, 977, 1133
<code>\@GLSuseriii</code> .. 158, 475, 727, 980, 1136	<code>\@Glsname</code> .. 151, 472, 724, 977, 1133
<code>\@GLSuseriii@</code> .. 158,	<code>\@Glsname@</code> .. 151,
375, 475, 573, 727, 826, 980, 1136	372, 472, 571, 724, 824, 977, 1133
<code>\@GLSuseriv</code> .. 159, 475, 727, 981, 1136	<code>\@Glspl</code> .. 144, 469, 721, 974, 1130
<code>\@GLSuseriv@</code> .. 159,	<code>\@Glspl@</code> .. 114, 117,
375, 475, 573, 727, 826, 981, 1136	144, 144, 348, 449, 450, 469,
<code>\@GLSuserv</code> .. 160, 476, 727, 981, 1137	470, 608, 700, 701, 721, 861,
<code>\@GLSuserv@</code> .. 160,	957, 958, 974, 1111, 1113, 1130
375, 476, 573, 727, 826, 981, 1137	<code>\@Glsplural</code> .. 149, 472, 723, 977, 1132
<code>\@GLSuservi</code> ..	<code>\@Glsplural@</code> .. 149,
161, 476, 728, 981, 982, 1137	372, 472, 571, 723, 824, 977, 1132
<code>\@GLSuservi@</code> .. 161,	<code>\@Glsymbol</code> .. 154, 474, 725, 979, 1134
376, 476, 573, 728, 827, 982, 1137	<code>\@Glsymbol@</code> .. 154,
<code>\@Gls</code> .. 141, 142, 468, 720, 973, 1129	373, 474, 572, 725, 825, 979, 1134
<code>\@Gls@</code> .. 113, 115,	<code>\@Glsymbolplural</code> ..
142, 142, 347, 348, 449, 450,	155, 474, 725, 979, 1135
468, 608, 699, 701, 720, 861,	<code>\@Glsymbolplural@</code> .. 155,
956, 958, 973, 1111, 1112, 1129	374, 474, 572, 725, 825, 979, 1135
<code>\@Gls@@link</code> .. 126, 963, 1117	<code>\@Glstext</code> 147, 471, 723, 976, 1131, 1132
<code>\@Gls@acentryname</code> ..	<code>\@Glstext@</code> .. 147,
171, 260, 481, 527,	371, 471, 570, 723, 823, 976, 1132
733, 778, 987, 1033, 1143, 1189	<code>\@Glsuser@i</code> 374–376, 572, 573, 825, 826
<code>\@Gls@entry@field</code> .. 170,	<code>\@Glsuseri</code> .. 156, 474, 726, 979, 1135
170, 171, 172, 436, 481–484,	<code>\@Glsuseri@</code> ..
687, 732–736, 986–988, 1142–1144	156, 374, 474, 726, 979, 1135
<code>\@Gls@entryname</code> ..	<code>\@Glsuserii</code> .. 157, 475, 726, 980, 1135
171, 171, 260, 481, 527,	<code>\@Glsuserii@</code> ..
733, 778, 987, 1033, 1143, 1189	157, 374, 475, 726, 980, 1135
<code>\@GlsSetXdyFirstLetterAfterDigits</code>	<code>\@Glsuseriii</code> .. 158, 475, 726, 980, 1136
183, 487, 738, 739, 993, 1149	<code>\@Glsuseriii@</code> ..
<code>\@GlsSetXdyNumberGroupOrder</code> ..	158, 375, 475, 726, 727, 980, 1136
184, 487, 739, 993, 994, 1149	<code>\@Glsuseriv</code> .. 159, 475, 727, 981, 1136
<code>\@Glsdesc</code> .. 152, 473, 724, 978, 1133	

<code>\@Glsuseriv@</code>	<code>\@backslashchar</code> 131, 132, 137, 138,
... 159, <u>375</u> , 475, 727, 981, 1136	192, 461, 466, 467, 712, 713,
<code>\@Glsuserv</code> .. 160, 476, 727, 981, 1137	718, 745, 965, 966, 971, 972,
<code>\@Glsuserv@</code>	1000, 1120, 1121, 1126, 1127, 1156
... 160, <u>375</u> , 476, 727, 981, 1137	<code>\@bsphack</code> .. 206, 503, 754, 1010, 1165
<code>\@Glsuservi</code> .. 161, 476, 728, 981, 1137	<code>\@cGls</code> .. 115, <u>115</u> , 450, 700, 957, 1112
<code>\@Glsuservi@</code>	<code>\@cGls@</code>
... 161, <u>376</u> , 476, 728, 981, 1137	113, 115, <u>115</u> , 449,
<code>\@Mi</code>	450, 699–701, 956–958, 1111, 1112
306, 619, 872	<code>\@cGlspl</code> 116, 450, 701, 958, 1113
<code>\@PGLS</code>	<code>\@cGlspl@</code> 114, 116, <u>116</u> , 449,
349, <u>349</u> , 608, 862	450, 700, 701, 956, 958, 1111, 1113
<code>\@PGLS@</code>	<code>\@cclv</code>
349, <u>349</u> , 608, 609, 862	306, 619, 872, 873
<code>\@PGLSpl</code>	<code>\@cgls</code> .. 115, <u>115</u> , 449, 700, 957, 1112
349, <u>349</u> , 609, 862	<code>\@cgls@</code>
<code>\@PGLSpl@</code>	113, 115, <u>115</u> , 448–
349, <u>349</u> , 609, 862	450, 699, 700, 956, 957, 1111, 1112
<code>\@Pgls</code>	<code>\@cglspl</code> 116, <u>116</u> , 450, 701, 958, 1113
347, <u>347</u> , 607, 860	<code>\@cglspl@</code> 113, 116, <u>116</u> , 448,
<code>\@Pgls@</code>	450, 699, 701, 956, 958, 1111, 1113
347, <u>347</u> , 607, 860	<code>\@chapter</code> 47, 411, 661, 916, 1071
<code>\@Pglspl</code>	<code>\@classoptionslist</code>
348, <u>348</u> , 608, 861 42, 410, 660, 915, 1070
<code>\@Pglspl@</code>	<code>\@closegls</code>
348, <u>348</u> , 608, 861	195, <u>195</u> , 495,
<code>\@Roman</code> .. 208, 211, 212, 504, 505,	496, 747, 748, 1002, 1157, 1158
755–757, 1011, 1012, 1166–1168	<code>\@colht</code>
<code>\@acrfull</code> 258, <u>258</u> , 525, 777, 1032, 1188	306, 619, 873
<code>\@acrfullpl</code>	<code>\@colroom</code>
259, 526, 778, 1032, 1033, 1188	306, 619, 873
<code>\@acrlong</code>	<code>\@currentlabelname</code>
165, 11, 242, 393, 517,
257, <u>370</u> , 479, 525, 569, 730,	642, 769, 897, 1025, 1051, 1180
777, 822, 984, 1032, 1140, 1187	<code>\@curroptions</code> . 42, 410, 660, 915, 1070
<code>\@acrlongpl</code> ... 167, 258, 480, 526,	<code>\@declaredoptions</code>
731, 778, 985, 1032, 1141, 1187 42, 410, 660, 915, 1070
<code>\@acrshort</code>	<code>\@delimN</code>
162, 524, 775
257, <u>369</u> , 476, 525, 569, 728,	<code>\@delimR</code>
777, 822, 982, 1032, 1137, 1187 523, 524, 775
<code>\@acrshortpl</code>	<code>\@disable@onlypremakeg</code>
163, 164, 258, 477, 526, 729, <u>47</u> , 199, 411, 498,
778, 983, 1032, 1138, 1187, 1188	661, 750, 917, 1005, 1071, 1161
<code>\@addtoacronymlists</code>	<code>\@disable@premakecs</code>
21, 398, 647, 648, 902, 1056, 1057 47, <u>47</u> , 411, 661, 917, 1071
<code>\@addtoacronymlists</code>	<code>\@disabled@glsaddxdycounters</code> ..
<u>21</u> <u>59</u> , 60, 417,
<code>\@afterheading</code> 294, 595, 613, 848, 866	418, 667, 668, 923, 924, 1078, 1079
<code>\@alph</code> 207, 211, 212, 503–505,	<code>\@do@addcounter</code>
755–757, 1010, 1012, 1166–1168	... 59, 417, 667, 923, 1077, 1078
<code>\@arabic</code> .. 208, 211, 212, 503–505,	<code>\@do@auxoutstuff</code> . 220, 221, 509,
755, 756, 1011, 1012, 1166–1168	510, 761, 762, 1017, 1172, 1173
<code>\@auxout</code>	<code>\@do@glossentry</code> 520, 553, 772, 806, 807
74, 76,	<code>\@do@gls@checkesc</code> 132, 713, 966, 1121
114, 138, 198, 201, 204, 216,	<code>\@do@gls@getcounterprefix</code>
221, 287, 426, 427, 449, 498,	213, 504, 505, 756, 757, 1013, 1169
499, 502, 507, 509, 510, 610,	
676, 677, 700, 749–751, 753,	
759, 761, 762, 863, 932, 933,	
957, 1004–1006, 1008, 1015,	
1017, 1046, 1087, 1088, 1112,	
1127, 1160, 1162, 1164, 1170, 1173	

<code>\do@gls@islistofacronyms</code>	1103, 1106, 1107, 1119, 1129–
. 21, 398, 648, 902, 1057	1131, 1137–1142, 1150, 1152,
<code>\do@glssee</code> 104, 444, 695, 951, 1106	1153, 1157–1159, 1165, 1169, 1184
<code>\do@ifinlist</code>	<code>\end@fixbraces</code>
. 58, 59, 417, 667, 922, 923, 1077 215, 507, 758, 759, 1014, 1170
<code>\do@newglossaryentry</code> 261, 527,	<code>\endfortrue</code>
537–548, 550, 582–587, 779, 31, 71, 91, 405, 424, 435,
789, 791–795, 797, 799–801,	610, 654, 674, 686, 863, 909,
803, 835–840, 1033, 1034, 1189	930, 942, 1046, 1063, 1085, 1096
<code>\do@seeglossary</code> 201, 214,	<code>\esphack</code> 207, 503, 755, 1010, 1166
215, 499, 506, 507, 751, 758,	<code>\expandtwoargs</code> 42, 410, 660, 915, 1070
759, 1006, 1014, 1162, 1169, 1170	<code>\@firstofone</code> 27, 35,
<code>\do@subglossentry</code> 521, 554, 773, 807	292, 401, 407, 650, 651, 656,
<code>\do@wrglossary</code>	866, 905, 911, 1059, 1060, 1066
128, 207, 459, 503, 589, 710,	<code>\@firstofthree</code> 125, 140,
755, 842, 964, 1010, 1118, 1166	141, 143, 146, 162, 164, 166,
<code>\do@write@glslabels</code> 36, 36,	168, 369–371, 458, 468–470,
37, 407, 656, 657, 911, 912, 1066	476, 478–480, 569, 570, 709,
<code>\do@writeaux@info</code>	719, 721, 722, 728–731, 822,
. 216, 507, 759, 1015, 1170	823, 962, 973–975, 982–985,
<code>\do@endpe</code> 220, 1172	1117, 1128–1131, 1137, 1139–1141
<code>\do@makeglossaries</code> 35,	<code>\@firstoftwo</code> 30, 31,
35, 198, 200, 406, 407, 498,	90, 91, 108, 125, 143–145, 164,
499, 656, 749, 751, 910, 911,	165, 168, 169, 403–405, 434,
1004, 1006, 1065, 1066, 1160, 1161	435, 458, 469, 470, 478, 480,
<code>\@ehc</code> 306, 619, 872	481, 653, 654, 685, 686, 709,
<code>\@empty</code> 14, 18, 21, 39, 41,	721, 722, 729–732, 907–909,
42, 47, 58, 60, 63, 66, 67, 100,	941, 942, 953, 962, 974, 975,
101, 106, 129, 141–145, 162–	983–986, 1062, 1063, 1096,
169, 185, 188, 190, 195–197,	1108, 1117, 1130, 1139, 1141, 1142
205, 206, 214, 250, 369–371,	<code>\@for</code> 31, 42, 47, 49, 59, 60,
394, 396, 398, 408–411, 417,	67, 89, 91, 131, 185, 187, 197–
418, 420, 422, 423, 441, 445,	199, 207, 215, 227, 228, 262,
460, 468–470, 476–481, 488,	405, 410, 411, 417, 418, 423,
490, 492, 495–497, 502, 503,	434, 435, 461, 488, 489, 497–
506, 522, 538, 539, 541–543,	499, 503, 507, 510, 511, 528,
545–547, 549, 550, 569, 570,	538, 540, 541, 543, 545, 547,
590, 592, 594, 644, 646, 647,	549, 551, 590, 610, 654, 660,
658–661, 667, 668, 670, 672,	661, 667, 668, 673, 685, 686,
673, 692, 696, 710, 711, 719–	712, 739–741, 749, 750, 755,
722, 728–732, 739, 741, 743,	759, 762, 780, 790, 792, 794,
747–749, 754, 758, 774, 790,	796, 798, 800, 802, 804, 843,
792–798, 800, 802, 803, 822,	863, 909, 915, 917, 923, 924,
823, 843, 845, 847, 898, 900,	928, 941, 942, 966, 994, 995,
902, 913, 915, 917, 923, 924,	1004, 1005, 1010, 1014, 1017,
926, 928, 948, 952, 964, 973–	1018, 1034, 1046, 1047, 1063,
975, 982–986, 994, 996, 998,	1070–1072, 1077, 1078, 1083,
1002, 1003, 1009, 1010, 1013,	1095, 1096, 1120, 1150, 1151,
1028, 1053, 1055, 1056, 1068–	1160, 1161, 1166, 1170, 1173, 1190
1071, 1077, 1078, 1081–1083,	<code>\@glo@desc</code> 103, 443, 694, 950, 1105

<code>\@glo@symbol</code>	103, 443, 694, 950, 1105	<code>\@glo@do@sortentries</code>
<code>\@glo@access</code> 352, 354, 355, 360, 554, 556, 559, 807, 809, 812	227, 228, 510, 511, 762, 763, 1017, 1018, 1173, 1174
<code>\@glo@addchildren</code> 227, 228, 232, 510, 511, 514, 762, 763, 766, 1018, 1021, 1173, 1174, 1177	<code>\@glo@entry</code>	.. 182, 486, 738, 993, 1148
<code>\@glo@assign@sortkey</code>	199, 201, 243, 499, 500, 518, 750, 751, 770, 1005, 1007, 1025, 1161, 1162, 1181	<code>\@glo@entryprefix</code> 343, 605, 858
<code>\@glo@autosee</code>	<code>\@glo@entryprefixfirst</code>
..... 104, 104, 444, 695, 951, 1105		343, 344, 605, 858
<code>\@glo@autoseehook</code>	<code>\@glo@entryprefixfirstplural</code>	..
..... 104, 104, 444, 695, 951, 1106		343, 344, 605, 858
<code>\@glo@check@mkidxrangear</code>	<code>\@glo@entryprefixplural</code>	343, 605, 858
..... 130, 213, 460, 461, 505, 589, 590, 711, 757, 842, 843, 965, 1012, 1120, 1168		<code>\@glo@esclabel</code>
<code>\@glo@check@sortallowed</code>	105, 106, 445, 696, 952, 1106, 1107
..... 16, 16, 17–19, 200, 203, 395–397, 499, 501, 645, 646, 750, 753, 899–901, 1005, 1008, 1054, 1055, 1161, 1164		<code>\@glo@etext</code> 451–453, 702, 703
<code>\@glo@childlist</code>	<code>\@glo@first</code>
..... 228, 511, 762, 1018, 1173		..	80, 99, 102, 103, 429, 440, 442, 443, 546, 586, 679, 691, 693, 694, 799, 839, 935, 947, 949, 950, 1089, 1102, 1104, 1105
<code>\@glo@counter</code> 81, 99, 102, 429, 440, 443, 679, 691, 693, 935, 947, 949, 1090, 1102, 1104	<code>\@glo@firstaccess</code>
<code>\@glo@counterprefix</code>	204, 209, 212– 214, 250, 253, 502, 504–506, 522, 524, 753, 756–758, 774, 776, 1009, 1011–1013, 1028, 1030, 1164, 1167–1169, 1184, 1186	352, 354, 355, 554, 556, 807, 809
<code>\@glo@default@sorttype</code>	<code>\@glo@firstplural</code>
..... 15, 15, 201, 230, 231, 395, 500, 512, 513, 645, 751, 764, 765, 899, 1007, 1019–1021, 1053, 1162, 1175, 1176		..	80, 99, 102, 429, 440, 442, 443, 586, 679, 691, 693, 839, 935, 947, 949, 1089, 1102, 1104
<code>\@glo@defaultcounter</code>	<code>\@glo@firstpluralaccess</code>	.. 352, 354, 355, 554, 556, 807, 809, 810
..... 102, 443, 693, 949, 1104		<code>\@glo@grabfirst</code>
<code>\@glo@desc</code> 79, 98– 100, 103, 429, 440, 441, 443, 679, 690–692, 694, 934, 946– 948, 950, 1089, 1101–1103, 1105	..	237, 237, 515, 766, 1022, 1177
<code>\@glo@descaccess</code>	<code>\@glo@label</code>	82, 84, 92–97, 99–104, 112, 180, 337, 338, 343, 344, 355, 356, 431, 436, 438–444, 448, 485, 486, 556, 557, 605, 637, 681, 686–695, 698, 737, 809, 810, 858, 891, 936, 937, 942–951, 955, 991, 992, 1091, 1092, 1097–1106, 1110, 1147, 1148
..... 352, 354, 355, 555–557, 808–810		<code>\@glo@list</code>	.. 104, 444, 695, 951, 1106
<code>\@glo@descplural</code> 79, 98, 99, 429, 440, 679, 690, 691, 934, 946, 947, 1089, 1101, 1102	<code>\@glo@long</code> 72, 84, 100, 102, 425, 431, 441, 443, 675, 681, 691, 694, 931, 937, 947, 950, 1085, 1092, 1102, 1104
<code>\@glo@descpluralaccess</code>	<code>\@glo@longaccess</code>
..... 352, 354, 355, 555–557, 808–810		353–355, 555–557, 808–810
		<code>\@glo@longpl</code> 84, 100, 103, 431, 441, 443, 537, 539, 540, 542, 544, 546, 548, 583, 681, 692, 694, 789–791, 793, 795, 797, 799, 801, 836, 937, 948, 950, 1092, 1102, 1104

<code>\@glo@longpluralaccess</code>	542, 544, 546, 548, 583, 586, 353, 355, 356, 555–557, 808–810
<code>\@glo@name</code>	16, 79, 85, 99, 101– 103, 395, 428, 432, 440, 442, 443, 645, 678, 682, 691, 693, 694, 899, 934, 938, 947, 949, 950, 1054, 1089, 1092, 1102–1104
<code>\@glo@no@assign@sortkey</code>	199, <u>243</u> , 499, 518, 750, 770, 1005, 1025, 1161, 1181
<code>\@glo@nonumberlist</code>	82, 83, 430, 680, 936, 1091
<code>\@glo@numfmt</code>	213, 506, 590, 757, 843, 1013, 1168, 1169
<code>\@glo@parent</code>	18, 82, 99, 101, 102, 106, 228, 229, 396, 430, 441, 442, 445, 511, 646, 680, 691–693, 696, 763, 900, 936, 947–949, 952, 1018, 1055, 1090, 1102–1104, 1106, 1107, 1174
<code>\@glo@plural</code>	80, 99, 101, 102, 429, 440, 442, 585, 679, 691, 693, 838, 935, 947, 949, 1089, 1102–1104
<code>\@glo@pluralaccess</code>	352, 354, 355, 554, 556, 807, 809
<code>\@glo@prefix</code>	12, 82, 99, 106, 130, 213, 393, 430, 441, 445, 460, 461, 505, 589, 590, 643, 680, 691, 696, 711, 712, 757, 842, 843, 897, 936, 947, 952, 965, 1013, 1052, 1090, 1091, 1102, 1106, 1107, 1120, 1168
<code>\@glo@range</code>	213, 505, 506, 589, 590, 757, 842, 843, 1013, 1168
<code>\@glo@see</code>	81, 99, 104, 429, 440, 444, 679, 691, 695, 935, 947, 951, 1090, 1102, 1105, 1106
<code>\@glo@seeautonumberlist</code>	12, <u>12</u> , 81, 393, 429, 643, 680, 897, 935, 1052, 1090
<code>\@glo@short</code>	72, 83, 100, 102, 425, 431, 441, 443, 586, 675, 681, 691, 694, 839, 931, 937, 947, 950, 1086, 1091, 1102, 1104
<code>\@glo@shortaccess</code>	352, 354, 355, 555– 557, 583–586, 808–810, 836–839
<code>\@glo@shortpl</code>	84, 100, 102, 431, 441, 443, 537, 539, 540,
	542, 544, 546, 548, 583, 586, 681, 691, 694, 789–791, 793, 795, 797, 799, 801, 836, 839, 937, 947, 950, 1092, 1102, 1104
<code>\@glo@shortpluralaccess</code>	353–355, 555–557, 808–810
<code>\@glo@sort</code>	16, 18, 25, 26, 79, 99, 102, 106, 395, 397, 400, 401, 429, 440, 442, 445, 645, 646, 650, 679, 691, 693, 696, 899, 900, 904, 905, 935, 947, 949, 952, 1054, 1055, 1059, 1089, 1102, 1104, 1106, 1107
<code>\@glo@sortedinsert</code>	228, 229, <u>229</u> , 511, 763, 1018, 1019, 1174
<code>\@glo@sortentries</code>	226, 227, <u>227</u> , 510, 512, 513, 762, 764, 765, 1017, 1019, 1020, 1173, 1175, 1176
<code>\@glo@sorthandler@case</code>	227, <u>230</u> , 512, 513, 764, 765, 1019, 1020, 1175, 1176
<code>\@glo@sorthandler@letter</code>	226, <u>229</u> , 512, 763, 764, 1019, 1020, 1174, 1175
<code>\@glo@sorthandler@nocase</code>	227, <u>230</u> , 512, 513, 764, 765, 1019, 1020, 1175, 1176
<code>\@glo@sorthandler@word</code>	226, <u>229</u> , 511, 512, 763, 764, 1019, 1174, 1175
<code>\@glo@sortinghandler</code>	227, 229, 510, 511, 762, 763, 1017, 1019, 1173, 1174
<code>\@glo@sortinglist</code>	227, 229, 231, 232, 510, 511, 513, 514, 762, 763, 765, 1017, 1018, 1021, 1173, 1174, 1176
<code>\@glo@sortmacro@case</code>	<u>231</u> , 513, 765, 1020, 1176
<code>\@glo@sortmacro@def</code>	<u>231</u> , 513, 765, 1021, 1176
<code>\@glo@sortmacro@def@do</code>	<u>232</u> , 514, 765, 1021, 1177
<code>\@glo@sortmacro@letter</code>	<u>230</u> , 512, 764, 1020, 1175
<code>\@glo@sortmacro@nocase</code>	<u>231</u> , 513, 765, 1020, 1176
<code>\@glo@sortmacro@standard</code>	<u>231</u> , 513, 764, 1020, 1175

<code>\@glo@sortmacro@use</code>	228, 232, 233, 236, 237, 241, 242, 260, 261, 393, 396, 429,
. 232 , 514, 766, 1021, 1177	
<code>\@glo@sortmacro@word</code>	440, 441, 443, 444, 486, 498,
. 230 , 512, 764, 1019, 1175	502, 508–511, 514, 517, 527, 538–541, 543, 545, 547, 549, 550, 609, 610, 642, 643, 646, 679, 691–695, 737, 738, 750, 754, 760–763, 765, 766, 769, 778, 790, 792–796, 798, 800, 802, 803, 863, 897, 900, 935, 947–951, 992, 993, 1005, 1009, 1016–1018, 1021, 1022, 1024, 1025, 1033, 1051, 1055, 1090, 1102–1105, 1148, 1160, 1164, 1165, 1171–1174, 1177, 1180, 1189
<code>\@glo@sorttype</code>	
201, 233, 236, 237, 243, 500, 514, 515, 518, 751, 766, 767, 770, 1007, 1021, 1022, 1025, 1162, 1177, 1178, 1181	
<code>\@glo@storeentry</code>	
16–18, 105 , 395, 396, 445, 645, 646, 696, 899, 900, 952, 1054, 1055, 1106	
<code>\@glo@suffi</code>	
130, 213, 460, 461, 506, 590, 711, 712, 757, 843, 965, 1013, 1120, 1168	
<code>\@glo@symbol</code>	
72, 80, 99, 103, 425, 429, 440, 443, 542, 546, 675, 679, 691, 694, 794, 799, 930, 931, 935, 947, 950, 1085, 1090, 1102, 1105	
<code>\@glo@symbolaccess</code>	
. 352, 354, 355, 554, 556, 557, 586, 807, 809, 810, 839	
<code>\@glo@symbolplural</code>	
80, 99, 103, 429, 440, 444, 679, 691, 694, 935, 947, 950, 1090, 1102, 1105	
<code>\@glo@symbolpluralaccess</code>	
352, 354, 355, 554, 556, 557, 808–810	
<code>\@glo@text</code>	
79, 99, 102, 103, 141–146, 170–172, 429, 440, 442, 443, 468–471, 481, 482, 542, 585, 606, 679, 691, 693, 694, 719–722, 732, 733, 795, 838, 859, 935, 947, 949, 950, 973–975, 987, 988, 1089, 1102, 1104, 1105, 1129–1131, 1142, 1143	
<code>\@glo@textaccess</code> 352, 354, 355, 554, 556, 583–585, 807, 809, 836–838	
<code>\@glo@thislabel</code>	
. 105, 444, 695, 951, 1106	
<code>\@glo@thislettergrp</code>	
226, 235, 237, 238, 515, 516, 766, 767, 1022, 1023, 1177, 1178	
<code>\@glo@thisvalue</code>	
73, 74, 425, 426, 675, 676, 931, 932, 1086	
<code>\@glo@tmp</code>	
75, 92, 93, 214, 436, 506, 686, 687, 758, 932, 942, 943, 1013, 1087, 1097, 1098, 1169	
<code>\@glo@type</code>	
11, 18, 80, 99, 100, 102–104, 181, 182, 198, 204, 205, 218–221, 227,	
<code>\@glo@types</code>	36, 67, 68, 74 , 76, 111, 182, 197, 198, 200, 283, 337, 407, 423, 426, 427, 447, 486, 497–499, 552, 637, 657, 673, 676, 677, 698, 738, 749, 750, 805, 890, 911, 928, 929, 932, 933, 955, 992, 993, 1004, 1005, 1046, 1066, 1083, 1087, 1088, 1110, 1148, 1159–1161, 1201
<code>\@glo@useri</code>	83, 99, 102, 430, 441, 443, 680, 691, 694, 936, 947, 950, 1091, 1102, 1104
<code>\@glo@useriaccess</code>	353, 355, 356, 555–557, 808–810
<code>\@glo@userii</code>	83, 99, 102, 430, 441, 443, 680, 691, 694, 936, 947, 950, 1091, 1102, 1104
<code>\@glo@useriiaccess</code>	353, 355, 356, 555–557, 808–810
<code>\@glo@useriii</code>	83, 99, 102, 431, 441, 443, 681, 691, 694, 937, 947, 950, 1091, 1102, 1104
<code>\@glo@useriiiaccess</code>	353, 355, 356, 555–557, 808–810
<code>\@glo@useriv</code>	83, 99, 102, 431, 441, 443, 681, 691, 694, 937, 947, 950, 1091, 1102, 1104
<code>\@glo@userivaccess</code>	353, 355, 356, 555–557, 808–810
<code>\@glo@userv</code>	83, 99, 102, 431, 441, 443, 681, 691, 694, 937, 947, 950, 1091, 1102, 1104
<code>\@glo@uservaccess</code>	353, 355, 356, 555–557, 808–810

<code>\@glo@uservi</code>	83, 99, 102, 431, 441, 443, 681, 691, 694, 937, 947, 950, 1091, 1102, 1104	<code>\@gls@assign@symbol@field</code>	25
<code>\@glo@userviaccess</code>	353, 355, 356, 555–557, 808–810	<code>\@gls@assign@symbolplural@field</code> 25	
<code>\@glodesc</code>	103, 443, 694, 950, 1105	<code>\@gls@automake</code>	195, 199, 495, 496, 499, 747, 750, 1002, 1005, 1157, 1158, 1161
<code>\@glo@list@</code>	100, 441, 692, 948, 1103	<code>\@gls@automake@immediate</code>	196, 197, 496, 497, 748, 749, 1003, 1004, 1158–1160
<code>\@glo@name</code>	103, 443, 694, 950, 1104	<code>\@gls@automake@types</code>	197, 197, 199, 1004, 1005, 1159–1161
<code>\@glossary@default@style</code>	11, 11, 13, 218, 250, 278, 393, 394, 508, 522, 523, 551, 643, 760, 774, 775, 804, 897, 898, 1016, 1029, 1044, 1051, 1052, 1171, 1184, 1200	<code>\@gls@before</code>	21, 398, 648, 902, 1057
<code>\@glossaryentryfield</code>	105, 106, 445, 695, 696, 951, 952, 1106, 1107	<code>\@gls@between</code>	288, 610, 863, 1047
<code>\@glossarysection</code>	55, 56, 415, 665, 920, 921, 1075, 1076	<code>\@gls@body</code> 172, 482, 733, 987, 988, 1143	
<code>\@glossarystyle</code>	218, 219, 241, 508, 509, 517, 760, 761, 769, 1016, 1024, 1171, 1172, 1180	<code>\@gls@checkactual</code>	131, 136, 192, 462, 465, 466, 493, 712, 716, 717, 745, 966, 970, 999, 1121, 1125, 1155
<code>\@glossarysubentryfield</code> 105, 106, 445, 695, 696, 951, 952, 1106, 1107		<code>\@gls@checkbox</code>	131, 135, 192, 462, 464, 465, 493, 713, 715, 716, 745, 966, 969, 1000, 1121, 1124, 1155
<code>\@gls</code>	141, 141, 468, 719, 973, 1128	<code>\@gls@checkedmidx</code>	130–138, 191–193, 461–467, 492–494, 712–718, 744–746, 965–972, 998–1001, 1120–1127, 1154–1156
<code>\@gls@</code>	113, 115, 141, 141, 346, 347, 448, 450, 468, 607, 608, 699, 700, 719, 860, 861, 956, 957, 973, 1111, 1112, 1128, 1129	<code>\@gls@checkesc</code>	132, 136, 192, 713, 745, 966, 1000, 1121, 1156
<code>\@gls@@automake@immediate</code>	197, 198, 497, 498, 749, 1004, 1159, 1160	<code>\@gls@checkescactual</code> 131, 133, 192, 462, 463, 493, 712, 714, 745, 966–968, 999, 1121–1123, 1155	
<code>\@gls@@link</code>	126, 126, 458, 709, 962, 963, 1117	<code>\@gls@checkescbar</code> 131, 134, 192, 462–464, 493, 713–715, 745, 966, 968, 1000, 1121, 1123, 1155	
<code>\@gls@Hcounter</code>	128, 129, 460, 710, 711, 964, 1119	<code>\@gls@checkesclevel</code> 132, 134, 192, 464, 713, 715, 745, 966, 968, 969, 1000, 1121, 1123, 1124, 1155	
<code>\@gls@ReturnAfterFi</code>	254, 524, 776, 1030, 1031, 1186	<code>\@gls@checkescquote</code>	131, 133, 191–193, 462, 463, 493, 494, 712–714, 745, 746, 966, 967, 999–1001, 1121, 1122, 1155, 1156
<code>\@gls@access@display</code>	360, 559, 812	<code>\@gls@checklevel</code>	132, 135, 192, 462, 465, 493, 713, 716, 745, 966, 969, 970, 1000, 1121, 1124, 1125, 1155
<code>\@gls@actualchar</code>	106, 133, 136, 189, 445, 463, 465, 466, 490, 491, 593, 696, 714, 716, 717, 742, 846, 952, 968, 970, 997, 1107, 1123, 1125, 1152	<code>\@gls@checkmidxchars</code>	105, 106, 130, 131, 191, 200, 212, 214, 445, 460, 461, 493, 499, 505, 506, 590, 696, 711, 712, 744, 751, 757, 758, 843, 952, 965, 966,
<code>\@gls@addpredefinedattributes</code>	61, 184, 193, 418, 419, 488, 495, 668, 669, 739, 746, 924, 994, 1001, 1079, 1149, 1157		
<code>\@gls@adjustmode</code>	181, 181, 486, 738, 992, 1148		
<code>\@gls@after</code>	21, 398, 648, 902, 1057		

999, 1006, 1012, 1014, 1106, 1120, 1121, 1155, 1162, 1168, 1169	410, 518, 644, 660, 770, 898, 916, 1025, 1026, 1053, 1071, 1181
<code>\@gls@checkquote</code>	<code>\@gls@define@glossarysubentrycounter</code>
. 131, 132, 191, 462, 492, 493, 712, 713, 744, 966, 967, 998, 999, 1121, 1122, 1154, 1155 15, 46, 242, 244, 395, 411, 518, 644, 661, 770, 898, 916, 1025, 1026, 1053, 1071, 1181
<code>\@gls@classI</code>	<code>\@gls@defsort</code>
185, 186, 488, 740, 994, 995, 1150	16–19, 103, 395–397, 443, 645, 646, 694, 899–901, 950, 1054, 1055, 1105
<code>\@gls@classII</code>	<code>\@gls@defsortcount</code>
. . . 185, 186, 488, 740, 995, 1150	16–19, 77, 395–397, 427, 645, 646, 677, 899–901, 933, 1054, 1055, 1088
<code>\@gls@codepage</code> . 66, 221, 422, 510, 672, 762, 928, 1017, 1082, 1173	<code>\@gls@deprecated@acr@warning</code>
<code>\@gls@counter</code> 277, 792, 793, 795, 797, 800, 802, 1044, 1199
123, 127, 129, 181, 204, 213, 214, 458–460, 486, 502, 506, 590, 708, 710, 711, 737, 738, 753, 757, 758, 843, 962–965, 992, 1009, 1013, 1014, 1116, 1118, 1119, 1148, 1164, 1168, 1169	<code>\@gls@do@acronymsdef</code>
<code>\@gls@counterwithin</code>	20, 20, 46, 78, 397, 398, 410, 411, 428, 647, 660, 661, 678, 901, 902, 916, 934, 1056, 1071, 1088
. 14, 15, 394, 395, 644, 898, 1053	<code>\@gls@do@automake@err</code>
<code>\@gls@ctr</code> 59, 417, 667, 923, 1077, 1078 37, 38, 912, 913, 1067, 1068
<code>\@gls@currentlettergroup</code>	<code>\@gls@do@glsunset</code>
. 236, 238, 514–516, 766, 767, 1021, 1023, 1177, 1178 124, 124, 141–146, 962, 973–975, 1117, 1129–1131
<code>\@gls@debugfalse</code> 7, 391, 640, 894, 1049	<code>\@gls@do@indexdef</code> 41, 42, 46, 78, 409–411, 428, 659–661, 678, 915, 916, 934, 1069–1071, 1088
<code>\@gls@debugtrue</code>	<code>\@gls@do@numbersdef</code> . . 41, 46, 78, 409–411, 428, 659–661, 678, 914, 916, 934, 1069, 1071, 1088
. 7, 390, 391, 640, 894, 1049	<code>\@gls@do@symbolsdef</code> . 41, 78, 409, 428, 659, 678, 914, 934, 1069, 1088
<code>\@gls@declareoption</code> 11, 12, 13, 20, 23, 24, 30, 33–38, 41, 42, 393, 394, 397–400, 404, 406, 407, 409, 410, 643, 647, 649, 653, 655–657, 659, 660, 897, 901, 903, 908, 910–915, 1051, 1052, 1056, 1058, 1062, 1064–1067, 1069	<code>\@gls@do@symbolssdef</code>
<code>\@gls@default</code> 117, 450, 701, 958, 1113	46, 410, 411, 660, 661, 916, 1071
<code>\@gls@default@value</code>	<code>\@gls@doautomake</code>
. 72–74, 80, 85, 86, 99, 101–103, 277, 343, 425, 426, 429, 432, 433, 440, 442, 443, 545, 605, 675, 676, 679, 682, 683, 691, 693, 694, 798, 858, 931, 932, 935, 938, 939, 947, 949, 950, 1044, 1085, 1086, 1089, 1092, 1093, 1102–1105, 1199 38, 38, 197, 199, 407, 408, 497, 499, 657, 658, 749, 750, 912, 913, 1004, 1005, 1067, 1068, 1160, 1161
<code>\@gls@defaulttype</code> . 74, 75, 932, 1087	<code>\@gls@docheckquotedef</code>
<code>\@gls@deffile</code> 191–193, 492–494, 744–746, 998–1001, 1154–1156
88–90, 433, 434, 683–685, 939–941, 1094–1096	<code>\@gls@docloadedfalse</code>
<code>\@gls@define@glossaryentrycounter</code> 14, 14, 46, 242, 244, 394, 395, 6, 390, 640, 894, 1048
	<code>\@gls@docloadedtrue</code>
	. 6, 390, 639, 640, 893, 894, 1048
	<code>\@gls@dodedeflistparser</code>
 199, 498, 750, 1005, 1161
	<code>\@gls@doentrycounterdef</code>
	46, 410, 411, 660, 661, 916, 1071
	<code>\@gls@doentrydef</code>
	457, 707, 708

<code>\@gls@dolast</code>	215, 507, 759, 1014, 1170	<code>\@gls@firsttok</code>
<code>\@gls@donext</code>	237, 514, 515, 766, 1022, 1177
.	215, 507, 759, 1014, 1015, 1170	<code>\@gls@fixbraces</code>	104, <u>215</u> , 444, 507, 695, 758, 951, 1014, 1106, 1170
<code>\@gls@donext@def</code>	<code>\@gls@forbidtexext</code>	.. <u>75</u> , 76, 426, 427, 676, 677, 932, 933, 1087, 1088
...	180, 485, 486, 737, 992, 1147	<code>\@gls@fulllist</code>	... 36, 657, 911, 1066
<code>\@gls@dosubentrycounterdef</code>	<code>\@gls@get@counterprefix</code>
.....	46, 411, 661, 916, 1071	214, 506, 758, 1013, 1169
<code>\@gls@dothiswrite</code> 495–497, 747–749, 1002–1004, 1158, 1159	<code>\@gls@getbody</code>	172, 482, 733, 987, 1143
<code>\@gls@elem</code> 610, 863, 1046	<code>\@gls@getcounter</code>
<code>\@gls@enablesavenonumberlist</code>	<u>78</u> , 428, 678, 934, 1088
.....	<u>82</u> , 89, 430, 434, 680, 684, 936, 940, 1091, 1095	<code>\@gls@getcounterprefix</code>	... 213, <u>213</u> , 504–506, 756–758, 1013, 1169
<code>\@gls@encapchar</code> 134, 135, 189, 213, 214, 464, 465, 490, 491, 506, 507, 590, 593, 715, 716, 742, 757, 758, 843, 846, 968, 969, 997, 1013, 1014, 1123, 1124, 1152, 1153, 1169, 1170	<code>\@gls@getgrouptitle</code>	201, 249, <u>249</u> , 288, 499, 521, 610, 751, 773, 863, 1006, 1028, 1047, 1162, 1184
<code>\@gls@entry@count</code>	<code>\@gls@glossary</code>	... 205, <u>205</u> , 206, 502, 503, 754, 1009, 1010, 1165
...	114, <u>115</u> , 449, 700, 957, 1112	<code>\@gls@glslink@hyper@update@hook</code>
<code>\@gls@entry@field</code>	124, <u>124</u> , 962, 1117
.....	92, 93, <u>112</u> , 139, <u>169</u> , 170, 171, 173–179, 356–358, 436, 448, 481–485, 557, 558, 686, 687, 699, 732–736, 810, 811, 942, 943, 956, 986–991, 1097, 1098, 1110, 1128, 1142–1147	<code>\@gls@gobbleopt</code>	75, 427, 677, 933, 1087
<code>\@gls@escbsdq</code> 130, 131, 190, 461, 492, 594, 712, 743, 847, 965, 966, 998, 1120, 1121, 1153	<code>\@gls@grptitle</code>
<code>\@gls@expand@fields</code>	249, 285, 288, 521, 609, 610, 773, 863, 1028, 1046, 1047, 1184
.....	85, <u>85</u> , 86, 432, 433, 682, 683, 938, 939, 1093, 1094	<code>\@gls@hyp@opt</code>	... 93, 94, 115, 116, <u>125</u> , 126, 141–169, 258, 259, 346–349, 437, 449, 450, 458, 468–480, 525, 526, 606–609, 687, 688, 700, 701, 709, 719– 732, 777, 778, 859–862, 943, 944, 957, 958, 962, 963, 973– 986, 1032, 1033, 1098, 1099, 1112, 1113, 1117, 1128–1142, 1188
<code>\@gls@expandonce</code>	<code>\@gls@hyp@opt@cs</code>
...	86, 433, 683, 939, 1093, 1094	125, 458, 709, 962, 1117
<code>\@gls@extramakeindexopts</code>	<code>\@gls@hypergroup</code>
.....	<u>193</u> , 198, 495, 498, 746, 749, 1001, 1005, 1157, 1160	..	287, <u>288</u> , 610, 863, 1046, 1047
<code>\@gls@fetch@lettergrp</code>	<code>\@gls@if@measuring</code>
.....	226, 227, 235, 238	107, <u>107</u> , 952, 953, 1107
<code>\@gls@fetchfield</code>	... 73, <u>91</u> , 426, 435, 676, 686, 931, 942, 1086, 1096	<code>\@gls@ifinlist</code>
<code>\@gls@field@link</code> 93, 94, <u>146</u> , 147–161, 371–376, 437, 471–476, 570–573, 687, 688, 722–728, 823–827, 943, 944, 975–982, 1098, 1099, 1131–1137	.	<u>58</u> , 59, 417, 667, 922, 923, 1077
<code>\@gls@fieldaccess@display</code>	. 360, <u>360</u> , 361–363, 559–561, 812–814	<code>\@gls@ifnotmeasuring</code> 446, 696
		<code>\@gls@igtype</code>	.. 79, 428, 678, 934, 1089
		<code>\@gls@increment@currcount</code>
		113, <u>114</u> , 448, 449, 699, 700, 956, 957, 1111, 1112
		<code>\@gls@indexdef</code> 41, <u>42</u> , 409, 410, 659, 660, 915, 1069, 1070

<code>\@gls@initnonnumberlist</code>	<code>\@gls@loadlong</code>
..... 82, <u>82</u> , 99, 430,	<u>12</u> , 13, 278, 393, 394, 551, 643,
441, 680, 691, 936, 947, 1091, 1102	804, 897, 898, 1044, 1052, 1200
<code>\@gls@islistofacronyms</code>	<code>\@gls@loadshellest</code>
..... 21, 398, 648, 902, 1057	43, 45
<code>\@gls@keylist</code>	<code>\@gls@loadsuper</code>
..... 387, 582, 835	12, <u>12</u> , 13, 278, 394, 551, 643,
<code>\@gls@keymap</code> ... 83, 89, <u>90</u> , 91, 92,	804, 897, 898, 1044, 1052, 1200
343, 354, 430, 434–436, 555,	<code>\@gls@loadtree</code>
605, 680, 685–687, 808, 858,	.. 13, <u>13</u> , 278, 394, 551, 643,
936, 941–943, 1091, 1095–1097	804, 897, 898, 1044, 1052, 1200
<code>\@gls@label</code>	<code>\@gls@local@increment@currcount</code>
201, 204, 210, 113, <u>114</u> , 448,
212, 213, 499, 502, 504–506,	449, 699, 700, 956, 957, 1111, 1112
751, 753, 756, 757, 1006, 1009,	<code>\@gls@loclist</code> . 202, 203, 234–236,
1012, 1013, 1162, 1164, 1167, 1168	238, 239, 500, 501, 515, 516,
<code>\@gls@labelsfile</code>	752, 753, 767, 768, 1007, 1008,
36, 407, 656, 657, 911, 912, 1066	1022, 1023, 1162, 1163, 1178, 1179
<code>\@gls@langmod</code> .. 34, 195, 196, 221,	<code>\@gls@map</code>
495, 496, 747, 748, 910, 1002,	89–91, 434,
1003, 1017, 1065, 1158, 1159, 1173	435, 685, 686, 941, 942, 1095, 1096
<code>\@gls@levelchar</code>	<code>\@gls@missinglang@warn</code>
106, 134, 135, 189, 445, 464, 23, <u>23</u> , 50, 399,
465, 490, 491, 593, 696, 715,	413, 649, 663, 903, 918, 1058, 1073
716, 742, 846, 952, 969, 970,	<code>\@gls@missingnumberlist</code>
997, 1107, 1123–1125, 1152, 1153 <u>84</u> , 103, 431,
<code>\@gls@link</code>	443, 681, 694, 937, 950, 1092, 1105
. 126, <u>127</u> , 141–146, 162–169,	<code>\@gls@noaccess</code>
369–371, 458, 459, 468–471, <u>354</u> , 360, 556, 559, 809, 812
477–481, 569, 570, 709, 710,	<code>\@gls@noexpand@field</code>
719–722, 728–732, 822, 823,	<u>85</u>
962, 963, 973–976, 982–986,	<code>\@gls@noexpand@fields</code> <u>85</u> , 87, 432,
1117, 1118, 1129–1131, 1138–1142	433, 682, 683, 938, 939, 1092, 1094
<code>\@gls@link@checkfirsthyper</code>	<code>\@gls@nohyperlist</code> 22, <u>22</u> , 78, 127,
..... <u>126</u> , 141–146,	399, 428, 459, 648, 678, 710,
458, 468–470, 709, 719–722,	903, 934, 963, 1057, 1089, 1118
963, 973–975, 1118, 1129–1131	<code>\@gls@noidx@do</code> 236, <u>237</u> , 514, 515,
<code>\@gls@link@label</code>	766, 767, 1021, 1022, 1177, 1178
127, 459,	<code>\@gls@noidx@getgrouptitle</code>
538, 544, 710, 791, 796, 963, 1118 201, <u>249</u> , 499, 522,
<code>\@gls@link@nocheckfirsthyper</code> ..	751, 774, 1006, 1028, 1162, 1184
.... 146, <u>146</u> , 162–169, 471,	<code>\@gls@noidx@sanitizesort</code>
476–481, 722, 728–732, 975, 25, 203, 400, 501,
976, 982–986, 1131, 1137–1142	650, 753, 904, 1008, 1059, 1164
<code>\@gls@link@opts</code>	<code>\@gls@noidx@setsanitizesort</code> 29,
127, 459,	<u>29</u> , 203, 402, 403, 501, 652,
538, 544, 710, 791, 796, 963, 1118	753, 906, 907, 1008, 1061, 1164
<code>\@gls@list</code> 610, 863, 1046, 1047	<code>\@gls@noidx@loclist@finalsep</code> ...
<code>\@gls@listsuffix</code> 202, 500, 752, 1007, 1163
..... 58, 417, 667, 923, 1077	<code>\@gls@noidx@loclist@prev</code>
<code>\@gls@loadlist</code> 202, 239, 500, 516, 752,
.. 13, <u>13</u> , 278, 394, 551, 643,	768, 1007, 1023, 1024, 1163, 1179
804, 897, 898, 1044, 1052, 1200	

<code>\@gls@noidxloclist@sep</code>	<code>\@gls@quotechar</code>
. 202, 239, 500, 516, 752,	136, 189–191, 193, 462–466,
768, 1007, 1023, 1024, 1163, 1179	490–494, 593, 713–717, 742,
<code>\@gls@noist@toolate</code> 188, 190, <u>194</u> ,	744, 746, 846, 967–970, 997–
997, 998, 1001, 1152, 1154, 1157	1001, 1122–1125, 1152–1154, 1156
<code>\@gls@noref@warn</code> 200, <u>204</u> , 233, 237,	<code>\@gls@reference</code>
499, 501, 514, 751, 753, 766,	201, 204, <u>240</u> ,
1006, 1008, 1022, 1161, 1164, 1177	499, 502, 516, 751, 753, 768,
<code>\@gls@notranslatorhook</code> <u>30</u>	1006, 1009, 1024, 1162, 1164, 1179
<code>\@gls@numberlink</code> 253,	<code>\@gls@removedoption</code>
524, 775, 776, 1030, 1185, 1186	<u>28</u> , 32, 33,
<code>\@gls@numbersdef</code>	39, 906, 909, 913, 1060, 1064, 1068
. 41, <u>41</u> , 409, 659, 914, 915, 1069	<code>\@gls@removespaces</code>
<code>\@gls@numlist@lastsep</code> 253, 254, 524, 776, 1030, 1186
. 180, 485, 486, 737, 992, 1147	<code>\@gls@renewglossary</code>
<code>\@gls@numlist@nextsep</code> 194, <u>206</u> , 495, 503,
. 180, 485, 486, 737, 992, 1147	747, 754, 1001, 1010, 1157, 1165
<code>\@gls@numlist@sep</code>	<code>\@gls@replacementtext</code>
180, 485, 486, 737, 991, 992, 1147 359, 360, 559, 812
<code>\@gls@old@chapter</code>	<code>\@gls@requested@languages</code>
. 47, 411, 661, 916, 1071 <u>32</u> ,
<code>\@gls@old@print@noidx@glossary</code> <u>236</u>	<u>32</u> , 49, 909, 917, 1063, 1064, 1072
<code>\@gls@oldnewglossaryentryposthook</code>	<code>\@gls@rest</code> 172, 482, 733, 987, 988, 1143
. 355, 556, 809	<code>\@gls@restore@glslocal</code>
<code>\@gls@oldnewglossaryentryprehook</code> <u>124</u> , 128, 962, 964, 1117, 1119
. 354, 556, 809	<code>\@gls@restoreat</code> 89, 434, 684, 940, 1095
<code>\@gls@onlypremakeg</code>	<code>\@gls@roman</code> 62, 419,
. 47, <u>47</u> , 411, 661, 917, 1071	590, 591, 669, 670, 844, 925, 1080
<code>\@gls@order</code> 195–197, 495–497,	<code>\@gls@run@output@dir</code> 38, <u>43</u> , 44
747–749, 1002–1004, 1157–1159	<code>\@gls@run@output@file</code> <u>43</u> , 45, 195–197
<code>\@gls@org@LT@output</code> 305, 619, 872	<code>\@gls@run@unrestricted@shell</code>
<code>\@gls@org@glsnoidxdisplayloc</code> 43, <u>43</u> , 195–197
. 203,	<code>\@gls@runshell</code> <u>43</u> , <u>43</u> , 44, 45, 195, 197
501, 752, 753, 1008, 1163, 1164	<code>\@gls@sanitized@tmp</code>
<code>\@gls@org@glsseeformat</code> 203, 131, 461, 712, 966, 1120
501, 752, 753, 1008, 1163, 1164	<code>\@gls@sanitizedesc</code>
<code>\@gls@override@glossary</code> <u>24</u> , 400, 405, 649, 655
. 40, <u>40</u> , 41, 408,	<code>\@gls@sanitizename</code>
409, 658, 659, 913, 914, 1068, 1069 <u>25</u> , 400, 650, 904, 1059
<code>\@gls@patchtabularx</code>	<code>\@gls@sanitizesort</code>
. 109, 446, 697, 953, 954, 1108 16, <u>25</u> , 395,
<code>\@gls@preglossaryhook</code>	400, 645, 650, 899, 904, 1054, 1059
. <u>218</u> , 220, 508, 509,	<code>\@gls@sanitizesymbol</code>
760, 761, 1016, 1017, 1171, 1172	<u>25</u> , 400, 405, 650, 655, 904, 1059
<code>\@gls@prevlevel</code> 317, 318, 338–	<code>\@gls@save@glslocal</code>
341, 600, 601, 625, 626, 638, 124, <u>124</u> , 128, 962, 963, 1117, 1118
639, 853, 854, 879, 880, 891, 892	<code>\@gls@saveentrycounter</code>
<code>\@gls@provide@newglossary</code> 128, <u>128</u> , 181, 459, 486, 710,
. <u>74</u> , 76, 426,	738, 964, 992, 1118, 1119, 1148
427, 676, 677, 932, 933, 1087, 1088	<code>\@gls@savenonumberlist</code>
 82, <u>82</u> , 83, 203, 430,
	680, 936, 1008, 1090, 1091, 1164

<code>\@gls@see@noindex</code>	9,	<code>\@gls@thislist</code>	
	9 , 81 , 392 , 430 , 641 , 642 , 680 ,		. . . 180 , 485 , 486 , 737 , 992 , 1147
	895 , 896 , 935 , 1050 , 1051 , 1090	<code>\@gls@thisloc</code> 214 , 506 , 758 , 1013 , 1169	
<code>\@gls@setacrstyle</code>	32 ,	<code>\@gls@thisval</code> . 91 , 435 , 686 , 942 , 1096	
	46 , 405 , 410 , 411 , 655 , 660 ,	<code>\@gls@title</code> . . . 55 , 414 , 664 , 920 , 1075	
	661 , 909 , 916 , 1064 , 1070 , 1071	<code>\@gls@tmp</code> . . . 8 , 18 , 50 , 62 , 63 , 86 ,	
<code>\@gls@setcounter</code>	77 ,	131 , 206 , 253 , 391 , 392 , 396 ,	
	77 , 427 , 428 , 677 , 678 , 933 , 1088	412 , 420 , 433 , 461 , 503 , 610 ,	
<code>\@gls@setdefault@glslink@opts</code> .		641 , 646 , 662 , 669 , 670 , 683 ,	
	. . . 127 , 128 , 459 , 710 , 963 , 1118	712 , 754 , 755 , 863 , 895 , 900 ,	
<code>\@gls@setsort</code>	16–19 ,	918 , 925 , 926 , 939 , 966 , 1010 ,	
	128 , 181 , 395–397 , 459 , 486 ,	1030 , 1047 , 1050 , 1055 , 1073 ,	
	645 , 646 , 710 , 738 , 899–901 ,	1080 , 1081 , 1094 , 1120 , 1165 , 1186	
	964 , 992 , 1054 , 1055 , 1118 , 1148	<code>\@gls@tmpb</code> . . . 132 , 132 , 133–138 ,	
<code>\@gls@setupshortcuts</code>	46 ,	191–193 , 462–467 , 492–494 ,	
	410 , 411 , 660 , 661 , 916 , 1070 , 1071	713–718 , 744–746 , 967–972 ,	
<code>\@gls@setupsort@clear</code> . 18 , 901 , 1055		998–1001 , 1121–1127 , 1154 , 1156	
<code>\@gls@setupsort@def</code>		<code>\@gls@toc</code>	56 , 57 , 57 ,
 17 , 396 , 645 , 899 , 1054		415 , 416 , 666 , 921 , 922 , 1076 , 1077
<code>\@gls@setupsort@none</code>		<code>\@gls@type</code> . . . 197 , 199 , 262 , 337 ,	
 18 , 397 , 646 , 900 , 1055	497 , 499 , 528 , 538 , 540 , 541 ,	
<code>\@gls@setupsort@standard</code>		543 , 545 , 547 , 549 , 551 , 637 ,	
 16 , 395 , 645 , 899 , 1054	749 , 750 , 780 , 790 , 792 , 794 ,	
<code>\@gls@setupsort@use</code>		796 , 798 , 800 , 802 , 804 , 891 ,	
 17 , 396 , 646 , 900 , 1054	1004 , 1005 , 1034 , 1160 , 1161 , 1190	
<code>\@gls@sort</code>	226 ,	<code>\@gls@updatechecked</code> 131 , 132 , 132 ,	
	227 , 235 , 238 , 515 , 767 , 1023 , 1178	191 , 192 , 461 , 462 , 493 , 494 ,	
<code>\@gls@sort@A</code>	229 , 230 , 511 ,	712 , 713 , 744 , 745 , 965–967 ,	
	512 , 763 , 764 , 1019 , 1174 , 1175	999 , 1000 , 1120 , 1121 , 1155 , 1156	
<code>\@gls@sort@B</code>	229 , 230 , 511 ,	<code>\@gls@usetranslator</code>	
	512 , 763 , 764 , 1019 , 1174 , 1175 30 , 30 , 31 , 32 , 49 ,	
<code>\@gls@startswitexpandonce</code>		403 , 404 , 412 , 653 , 654 , 662 ,	
 86 , 86 ,	907–909 , 918 , 1062–1064 , 1072	
	432 , 433 , 682 , 683 , 938 , 939 , 1093	<code>\@gls@value</code>	
<code>\@gls@storenonumberlist</code> 85 , 86 , 175 , 432 , 483 , 682 ,	
 82 , 83 , 103 , 430 ,	735 , 938 , 989 , 1092 , 1093 , 1145	
	443 , 680 , 694 , 936 , 950 , 1091 , 1105	<code>\@gls@warnonglossdefined</code>	
<code>\@gls@symbolsdef</code> 24 , 24 , 217 , 400 , 508 , 649 ,	
 41 , 41 , 409 , 659 , 914 , 1069	759 , 903 , 904 , 1015 , 1058 , 1171	
<code>\@gls@this</code> . 207 , 503 , 755 , 1010 , 1166		<code>\@gls@warnontheGLOSSdefined</code> . . .	
<code>\@gls@thisHloc</code> 24 , 24 , 245 , 400 , 520 , 649 ,	
 214 , 506 , 758 , 1013 , 1169	771 , 903 , 904 , 1027 , 1058 , 1183	
<code>\@gls@thisentryname</code> 36 , 657 , 911 , 1066		<code>\@gls@write@entrycounts</code>	
<code>\@gls@thisfield</code>		114 , 114 , 449 , 700 , 957 , 1111 , 1112	
 73 , 426 , 676 , 931 , 932 , 1086	<code>\@gls@writedef</code>	88 , 89 ,
<code>\@gls@thislabel</code>		89 , 434 , 684 , 685 , 940 , 941 , 1095	
 71 , 215 , 231 , 424 , 507 ,	<code>\@gls@writeisthook</code>	
	513 , 514 , 674 , 759 , 765 , 930 , 188 , 190 , 190 , 490 ,	
	1014 , 1021 , 1084 , 1085 , 1170 , 1176	492 , 742 , 743 , 997 , 998 , 1152 , 1154	
<code>\@gls@thislang</code>	49 , 917 , 1072	<code>\@gls@x</code>	253 , 776 , 1030 , 1186

<code>\@gls@xdy@Lclass@Alpha-page-numbers</code>	<code>\@glsdescplural</code>	152, 153, 473, 725, 978, 1134
.....	63	
<code>\@gls@xdy@Lclass@Appendix-page-numbers</code>	<code>\@glsdescplural@</code>	153,
.....	63	373, 473, 572, 725, 825, 978, 1134
<code>\@gls@xdy@Lclass@Roman-page-numbers</code>	<code>\@glsdisp</code>	
.....	62	145, 145, 146, 470, 722, 975, 1131
<code>\@gls@xdy@Lclass@alpha-page-numbers</code>	<code>\@glsentry</code>	36, 111, 114,
.....	62	407, 447, 449, 657, 698, 700,
<code>\@gls@xdy@Lclass@arabic-page-numbers</code>	62	911, 955, 957, 1066, 1110, 1112
.....	<code>\@glsentrytitlecase</code>	
<code>\@gls@xdy@Lclass@arabic-section-numbers</code>	63	175, 175, 483, 735, 989, 1145
.....	<code>\@glsfirst</code>	..	148, 471, 723, 976, 1132
<code>\@gls@xdy@Lclass@roman-page-numbers</code>	<code>\@glsfirst@</code>	148,
.....	62	371, 471, 570, 723, 824, 976, 1132
<code>\@gls@xdy@locationlist</code>	<code>\@glsfirstletter</code>	183, 183,
.....	62, 185, 419, 488,	184, 487, 738, 739, 993, 1148, 1149
669, 740, 925, 994, 995, 1079, 1150	<code>\@glsfirstplural</code>	
<code>\@gls@xdycheckbackslash</code>	131,	149, 150, 472, 723, 977, 1132
.....	<code>\@glsfirstplural@</code>	137, 461, 466, 467, 712, 718,	150, 372, 372,
.....	965, 971, 972, 1120, 1126, 1127	472, 571, 723, 724, 824, 977, 1132
<code>\@gls@xdycheckquote</code>	<code>\@glsnumber</code>	252, 252,
.....	131, 137, 461, 466,	523, 524, 775, 1029, 1030, 1185
.....	<code>\@glsisacronymlistfalse</code>	712, 717, 718, 965, 971, 1120, 1126
<code>\@gls@xref</code>	214,	22, 398, 648, 902, 1057
.....	<code>\@glsisacronymlisttrue</code>	506, 507, 758, 1014, 1169, 1170
<code>\@gls@alphacompositor</code>	214,	22, 398, 648, 902, 1057
.....	<code>\@glslink</code>	52, 53, 63, 414, 420, 591,	108,
.....	664, 670, 844, 919, 926, 1074, 1080	128, 139, 140, 180, 285, 459,
<code>\@gls@Hlocref</code>	..	209, 212, 504, 505,	467, 468, 486, 609, 710, 719,
.....	756, 757, 1011, 1012, 1167, 1168	737, 863, 953, 964, 972, 973,
<code>\@gls@acronymlists</code>	21, 21, 22, 67, 260,	992, 1046, 1108, 1118, 1128, 1148
.....	<code>\@glslocalreset</code>	262, 283, 398, 423, 527, 528,	109, 110, 113, 446–
.....	538–541, 543, 545, 547, 549–	448, 697–699, 954–956, 1109, 1111
.....	<code>\@glslocalunset</code>	552, 647, 648, 673, 778, 780,	110, 110, 113, 447,
.....	790, 792–798, 800, 802–805,	448, 697, 699, 954, 956, 1109, 1111
.....	<code>\@glslocref</code>	902, 928, 1033, 1034, 1046,
.....	1056, 1057, 1083, 1189, 1190, 1201	204, 209, 212, 213, 502, 504–
<code>\@gls@saddkey</code>	92, 436, 686, 687, 942, 943, 1097	506, 590, 753, 756, 757, 843,
.....	<code>\@glsminrange</code>	1009, 1011–1013, 1164, 1167–1169
<code>\@gls@saddstoragekey</code>	91, 435, 686, 942, 1097	184,
.....	59, 60, 418, 668, 923, 924, 1078	184, 185, 186, 487–489, 591,
<code>\@gls@saddxdyattribute</code>	59, 60, 418, 668, 923, 924, 1078	739, 740, 844, 994, 995, 1149, 1150
.....	<code>\@glsname</code>	150, 151, 472, 724, 977, 1133	150, 151, 472, 724, 977, 1133
<code>\@gls@defaultplural</code>	84, 431, 681
.....	<code>\@glsname@</code>	151,
<code>\@gls@defaultsort</code>	16, 85, 395,	372, 472, 571, 724, 824, 977, 1133
.....	<code>\@glsnavhypertarget</code>
<code>\@gls@desc</code>	151, 152, 473, 724, 978, 1133	287, 287, 609, 610, 863, 1046
.....	<code>\@glsnextpages</code>	152,	219, 243, 509, 518,
<code>\@gls@desc@</code>	373, 473, 571, 724, 824, 978, 1133	761, 770, 1016, 1025, 1172, 1181

<code>\@glsnodesc</code>	84, 99, 100, 103, 431, 440, 441, 443, 681, 691, 692, 694, 937, 947, 948, 950, 1092, 1102, 1103, 1105	<code>\@glssymbolplural@</code>	155, 374, 474, 572, 725, 825, 979, 1134
<code>\@glsnoname</code>	84, 99, 101, 103, 431, 440, 442, 443, 681, 691, 693, 694, 937, 947, 949, 950, 1092, 1102–1104	<code>\@glstarget</code> 140, <u>140</u> , 246, 287, 467, 468, 520, 610, 719, 772, 863, 972, 973, 1027, 1046, 1128, 1183	<code>\@glstext</code> . . . 147, 471, 722, 976, 1131
<code>\@glsnonextpages</code> 219, <u>243</u> , 509, 518, 761, 770, 1016, 1025, 1172, 1181		<code>\@glstext@</code> 147, 371, 471, 570, 722, 823, 976, 1131	<code>\@glssunset</code> . . . 110, <u>110</u> , 113, 446– 448, 697–699, 954, 956, 1109, 1111
<code>\@glsnumberformat</code> 124, 127, 181, 204, 213, 458, 459, 486, 502, 505, 506, 589, 590, 708, 710, 737, 738, 753, 757, 842, 843, 962, 963, 992, 1009, 1012, 1013, 1117, 1118, 1148, 1164, 1168		<code>\@glssuseri</code> 155, 156, 474, 726, 979, 1135	<code>\@glssuseri@</code> 156, 374, 474, 572, 726, 825, 979, 1135
<code>\@glsopenfile</code>	194, <u>194</u> , 205, 495, 502, 747, 754, 1001, 1002, 1009, 1157, 1165	<code>\@glssuserii</code> 156, 157, 474, 726, 980, 1135
<code>\@glsorder</code> 33, 198, 405, 498, 655, 749, 750, 910, 1004, 1005, 1064, 1160	<code>\@glssuserii@</code> 157, 374, 475, 572, 726, 826, 980, 1135	<code>\@glssuseriii</code>
<code>\@glspl</code> 143, 469, 721, 974, 1130		<code>\@glssuseriii@</code> 158, 375, 475, 573, 726, 826, 980, 1136	<code>\@glssuseriv</code>
<code>\@glspl@</code> 113, 116, 143, <u>143</u> , 347, 348, 449, 450, 469, 607, 608, 699, 701, 721, 860, 861, 956, 958, 974, 1111, 1113, 1130		<code>\@glssuseriv@</code> 158, 159, 475, 727, 980, 1136	<code>\@glssuseriv@</code> 159, 375, 475, 573, 727, 826, 980, 1136
<code>\@glsplural</code> 148, 149, 472, 723, 976, 1132	<code>\@glssuseriv@</code> 159, 160, 476, 727, 981, 1136	<code>\@glssuseriv@</code> 160, 375, 476, 573, 727, 826, 981, 1136
<code>\@glsplural@</code> 149, 372, 472, 571, 723, 824, 976, 1132		<code>\@glssuservi</code> 160, 161, 476, 727, 981, 1137
<code>\@glsprefix@record@hook</code> 345, 346–349, 606–609, 859–862	<code>\@glssuservi@</code> 161, 376, 476, 573, 727, 826, 981, 1137	<code>\@glsswidestname</code> 338, <u>338</u> , 339, 600, 601, 637, 638, 853, 854, 891, 892
<code>\@glsreset</code> . . . 109, <u>110</u> , 113, 446– 448, 697–699, 954–956, 1109, 1111		<code>\@glsswritefiles</code> 39, <u>204</u> , 408, 502, 658, 754, 913, 1009, 1068, 1164	<code>\@glssxtr@doaccsupp</code> 350, 553, 806
<code>\@glssee</code> 104, 215, 444, 507, 695, 759, 951, 1014, 1106, 1170		<code>\@glssxtr@record</code> 345, 606, 859	<code>\@gobble</code> . 7, 16–18, 35, 36, 89, 90, 108, 131, 183, 187, 199, 200, 292, 390, 391, 395–397, 406, 407, 434, 446, 461, 487, 489, 498, 499, 588, 592, 593, 640, 645, 646, 656, 657, 685, 697, 712, 738, 741, 750, 751, 842, 845, 846, 865, 894, 899, 900, 910–912, 941, 953, 966, 993, 995, 1005, 1006, 1049, 1054, 1055, 1065, 1066, 1095, 1096, 1108, 1120, 1148, 1151, 1160, 1162
<code>\@glsshowaccsupp</code> 7, <u>9</u> , 351, 391, 392, 554, 640, 641, 807, 894, 895, 1049, 1050		
<code>\@glsshowtarget</code> 7, 8, <u>8</u> , 390, 391, 467, 640, 641, 718, 719, 894, 895, 1049, 1050		
<code>\@glssymbol</code> 153, 154, 473, 725, 978, 1134		
<code>\@glssymbol@</code> 154, 373, 473, 572, 725, 825, 978, 1134			
<code>\@glssymbolplural</code> 154, 155, 474, 725, 979, 1134		

<code>\@gobblethree</code>	. 7, 391, 640, 894, 1049	<code>\@newglossary</code>	. . . 74, 76, 77, 426–428, 676–678, 932, 933, 1087, 1088
<code>\@idxitem</code> 332, 634, 888	<code>\@newglossaryentry@defcounters</code>	104, 104, 112, 444, 448, 695, 698, 951, 955, 1105, 1106, 1110
<code>\@ifclassloaded</code> 6, 14, 55, 390, 394, 415, 640, 644, 665, 894, 898, 920, 1048, 1052, 1075	<code>\@newglossaryentryposthook</code> 92, 93, 104, 104, 343, 355, 436, 444, 556, 605, 686, 687, 695, 809, 858, 942, 943, 951, 1097, 1105, 1106
<code>\@ifglossaryexists</code> 68, 68, 217, 423, 508, 673, 759, 929, 1015, 1083, 1171	<code>\@newglossaryentryprehook</code> 92, 93, 98, 100, 104, 343, 354, 436, 440, 441, 444, 556, 605, 686, 687, 691, 692, 695, 809, 858, 942, 943, 947, 948, 951, 1097, 1101, 1102, 1106
<code>\@ifl@t@r</code> 27, 402, 651, 905, 1060	<code>\@nil</code> 21, 104, 130–132, 172, 191, 192, 213, 215, 227, 237, 252–254, 398, 444, 460–462, 482, 493, 505, 507, 514, 515, 523, 524, 589, 590, 648, 695, 711–713, 733, 744, 745, 757, 758, 766, 767, 775, 776, 842, 843, 902, 951, 965–967, 987, 999, 1000, 1012, 1014, 1022, 1023, 1029, 1030, 1057, 1106, 1120, 1121, 1143, 1155, 1156, 1168, 1170, 1177, 1178, 1185, 1186
<code>\@ifnextchar</code> 77, 125, 427, 458, 677, 709, 933, 962, 1088, 1117	<code>\@nnil</code> 21, 215, 398, 507, 648, 759, 902, 1014, 1057, 1170
<code>\@ifpackageloaded</code> 6, 11, 12, 30, 31, 49, 66, 107, 179, 190, 350, 390, 393, 403–405, 412, 422, 446, 485, 492, 553, 639, 643, 653, 654, 662, 672, 696, 736, 744, 806, 893, 897, 907–909, 918, 928, 953, 991, 998, 1048, 1051, 1052, 1062, 1063, 1072, 1083, 1107, 1147, 1154	<code>\@no@makeglossaries</code>	35, 199, 201, 204, 406, 498, 500, 501, 656, 750, 751, 753, 910, 911, 1005, 1007, 1008, 1065, 1161, 1162, 1164
<code>\@ifstar</code> 68, 76, 91, 92, 125, 183, 184, 255, 423, 427, 435, 436, 458, 487, 525, 673, 677, 686, 709, 738, 739, 776, 929, 933, 942, 962, 993, 1031, 1083, 1087, 1097, 1117, 1149, 1187	<code>\@no@post@desc</code> 51, 413, 594, 663, 847, 848, 919, 1074
<code>\@ifundefined</code> 49, 296, 307, 318, 326, 339, 412, 600, 601, 610, 614, 620, 626, 627, 630, 631, 638, 662, 853, 854, 863, 867, 873, 880, 884, 892, 917, 1072	<code>\@nopostdesc</code>	. . . 51, 219, 413, 509, 663, 761, 919, 1016, 1074, 1172
<code>\@ignored@glossaries</code> 36, 78, 78, 79, 407, 428, 657, 678, 911, 934, 1066, 1088, 1089	<code>\@onelevel@sanitize</code>	. . . 8, 25, 36, 62, 90, 106, 130, 131, 183, 184, 188, 214, 216, 237, 253, 391, 400, 401, 419, 434, 445, 461, 487, 490, 506, 507, 515, 590–592, 641, 650, 657, 669, 685, 696, 712, 738, 739, 741, 758, 759, 766, 844, 845, 895, 904, 911, 925, 941, 952, 965, 966, 993, 996, 1014, 1015, 1022, 1030, 1050, 1059, 1066, 1080, 1095, 1107, 1120,
<code>\@input@</code>	. . . 220, 509, 761, 1017, 1172		
<code>\@istfilename</code> 52, 198, 414, 498, 664, 749, 750, 919, 1004, 1005, 1074, 1160		
<code>\@makecol</code> 306, 619, 873		
<code>\@makeglossaries@warn@noprintglossary</code> 199, 200, 750, 1005, 1161		
<code>\@makeglossary</code> 35, 194, 198, 199, 406, 407, 495, 498, 656, 746, 750, 910, 911, 1001, 1005, 1065, 1157, 1160		
<code>\@minus</code> 291, 312, 332, 612, 623, 634, 865, 876, 887		
<code>\@mkboth</code> 55, 56, 415, 665, 921, 1075, 1076		
<code>\@ne</code> 107, 953, 1107		

1149, 1152, 1169, 1170, 1177, 1186
 \@onlypreamble 77, 87, 98, 114, 117, 200,
 203, 428, 433, 440, 449, 451,
 499, 501, 678, 683, 691, 700,
 701, 750, 753, 933, 939, 947,
 957, 958, 1005, 1008, 1088,
 1094, 1102, 1112, 1113, 1161, 1164
 \@onlypremakeg 47, 51–53, 59, 60, 64, 77,
 190, 411, 414, 417, 418, 421,
 428, 492, 661, 664, 667, 668,
 671, 678, 743, 917, 919, 920,
 923, 924, 926, 933, 998, 1071,
 1074, 1078, 1079, 1081, 1088, 1154
 \@org@Gls@acrentryname 172, 987, 1143
 \@org@glossaryentrynumbers 219, 220,
 509, 760, 761, 1016, 1017, 1172
 \@org@Gls@assign@descplural 537, 544,
 545, 547–549, 583, 585–587,
 789, 797, 799–801, 836, 839, 840
 \@org@Gls@assign@firstpl 537–545,
 547–549, 583–587, 789, 791,
 793, 795, 797, 799–801, 836–840
 \@org@Gls@assign@plural 537–539, 541–545,
 547–549, 583–587, 789, 791,
 793, 795, 797, 799–801, 836–840
 \@org@Gls@assign@symbolplural 538, 539, 541–
 543, 547–549, 583–587, 789,
 791–793, 795, 799–801, 836–840
 \@org@Gls@numberformat 180, 485, 737, 991, 1147
 \@org@newglossaryentryprehook 98, 440, 691, 947, 1101
 \@outputpage 306, 619, 873
 \@p@glossarysection . 55, 56, 415,
 416, 665, 666, 920, 921, 1075, 1076
 \@pgls 346, 346, 606, 859
 \@pgls@ 346, 346, 606, 860
 \@pglspl 346, 346, 607, 860
 \@pglspl@ 346, 346, 607, 860
 \@plus 291, 312,
 332, 612, 623, 634, 865, 876, 887
 \@print@glossary 217, 220, 508, 509,
 760, 761, 1015, 1017, 1171, 1172
 \@print@noidx@glossary 218, 232, 508, 514,
 760, 766, 1015, 1021, 1171, 1177
 \@printgloss@checkexists 217, 217, 218, 219, 508,
 509, 760, 1015, 1016, 1171, 1172
 \@printgloss@checkexists@allowignored 217, 508, 760, 1015, 1171
 \@printgloss@checkexists@noignored 217,
 217, 218, 508, 759, 760, 1015, 1171
 \@printgloss@setsort 199, 201, 218, 219, 498, 500,
 508, 509, 750, 751, 760, 1005,
 1007, 1016, 1161, 1162, 1171, 1172
 \@printglossary 217, 218,
 218, 508, 760, 1015, 1016, 1171
 \@roman 62, 419, 590, 669, 844, 925, 1080
 \@secondofthree 125, 140, 142, 144, 162,
 164, 166, 168, 369, 458, 468,
 470, 477–480, 569, 709, 719–
 721, 728, 729, 731, 732, 822,
 962, 973, 975, 982, 983, 985,
 986, 1117, 1128–1130, 1138–1141
 \@secondoftwo 27, 30–32, 50,
 90, 91, 108, 140–143, 146, 162,
 163, 166, 167, 209, 292, 341,
 342, 369–371, 401, 403–405,
 412, 434, 435, 467–470, 476,
 477, 479, 569, 570, 588, 604,
 650, 651, 653, 654, 662, 685,
 686, 719, 720, 722, 728–731,
 822, 823, 841, 857, 865, 866,
 905, 907–909, 918, 941, 942,
 953, 972–975, 982–985, 1011,
 1059, 1060, 1062–1064, 1072,
 1095, 1096, 1108, 1128, 1129,
 1131, 1137, 1138, 1140, 1141, 1167
 \@set@glo@numformat 130, 213, 460, 506, 590, 711,
 757, 843, 965, 1013, 1120, 1168
 \@sglsaddkey . . 92, 436, 686, 942, 1097
 \@sglsaddstoragekey 91, 435, 686, 942, 1097
 \@tabacckludge 27, 401, 650, 905, 1059
 \@text@composite@x 27, 401, 650, 905, 1059
 \@thirdofthree . . . 125, 143, 145,
 163, 165, 167, 169, 369, 458,

469, 470, 477–479, 481, 569,
 709, 720, 722, 729–732, 822,
 962, 974, 975, 983–986, 1117,
 1129, 1130, 1138, 1139, 1141, 1142
 \@this@attr .. 187, 489, 741, 995, 1151
 \@this@childlabel 228,
 511, 762, 763, 1018, 1173, 1174
 \@this@counter 60, 418, 668, 924, 1078
 \@this@ctr .. 187, 489, 741, 995, 1151
 \@this@key ... 91, 435, 686, 942, 1096
 \@this@label
 . 227, 510, 762, 1017, 1018, 1173
 \@thiscs 47, 411, 661, 917, 1071
 \@tmp 419, 590, 591, 844
 \@use@option ... 42, 410, 660, 915, 1070
 \@warn@nomakeglossaries
 197, 221, 497, 510,
 749, 762, 1004, 1017, 1159, 1173
 \@wrglossary@pageformat 208, 209,
 504, 755, 756, 1011, 1166, 1167
 \@wrglossarynumberhook
 208, 208, 212, 504, 505,
 755–757, 1011, 1012, 1166–1168
 \@write@glslabelsname
 36, 37, 657, 912, 1066
 \@xdy@main@language
 .. 34, 34, 406, 495, 496, 510,
 655, 747, 748, 761, 910, 1064, 1065
 \@xdy@attributelist .. 58, 60, 187,
 417, 418, 489, 667, 668, 741,
 922, 924, 995, 1077, 1078, 1151
 \@xdy@attributes 58,
 60, 185, 417, 418, 488, 588,
 590, 667, 668, 740, 841, 843,
 922–924, 994, 1077, 1078, 1150
 \@xdy@counters 59, 60, 187,
 417, 418, 489, 667, 668, 741,
 923, 924, 995, 1077, 1078, 1151
 \@xdy@crossrefhook . 63, 186, 420,
 489, 670, 740, 926, 995, 1080, 1151
 \@xdy@language 65, 221, 422, 509, 510,
 672, 761, 927, 1017, 1082, 1173
 \@xdy@lettergroups 66, 67,
 188, 422, 423, 490, 593, 672,
 673, 742, 846, 928, 996, 1083, 1152
 \@xdy@locationclassorder ... 64,
 64, 186, 421, 489, 592, 671,
 740, 845, 926, 927, 995, 1081, 1151
 \@xdy@locoref 58,
 60, 188, 417, 418, 490, 588,
 592, 667, 668, 742, 841, 845,
 922, 923, 996, 1077, 1078, 1152
 \@xdynumbergrouporder
 67, 184, 184, 422,
 487, 672, 739, 928, 993, 1083, 1149
 \@xdy@requiredstyles 65, 65,
 185, 421, 422, 488, 590, 671,
 672, 739, 843, 927, 994, 1082, 1150
 \@xdy@sortrules
 .. 65, 65, 188, 421, 490, 593,
 671, 742, 846, 927, 997, 1082, 1152
 \@xdy@style
 185, 488, 590, 739, 843, 994, 1150
 \@xdy@useralphabets
 .. 61, 61, 185, 419, 488, 590,
 669, 740, 843, 925, 994, 1079, 1150
 \@xdy@userlocationdefs
 63, 63, 64, 186, 420,
 489, 589, 591, 670, 740, 842,
 844, 926, 995, 1080, 1081, 1151
 \@xdy@userlocationnames
 63, 64, 420, 421, 589,
 670, 671, 842, 926, 927, 1080, 1081
 \@xfor@nextelement
 215, 507, 759, 1014, 1170
 \@xp@gl@getcounterprefix . 209,
 212, 213, 1011–1013, 1167–1169
 \@xp@gl@shypernumber
 252, 252, 1029, 1185
 \\ 105, 131, 136, 183,
 189, 190, 252, 253, 445, 461,
 487, 491, 523, 524, 593, 594,
 596, 597, 603, 604, 695, 712,
 717, 738, 742, 743, 775, 776,
 846, 847, 849, 850, 856, 857,
 951, 966, 970, 971, 993, 997,
 998, 1029, 1030, 1106, 1120,
 1125, 1126, 1148, 1153, 1185, 1186
 \^ 27, 36,
 401, 651, 657, 905, 912, 1060, 1066
 _ ... 222–227, 232–234, 236, 240, 241
 \^ 27, 401, 651, 905, 1060
 \| 131, 134, 136, 192, 462–464, 493,
 713–715, 717, 745, 966, 968,
 970, 1000, 1121, 1123, 1125, 1155
 \~ 27, 401, 651, 905, 1060

 A
 \A 249, 1028, 1184
 \a 27, 401, 651, 905, 1060

\AA	27, 401, 651, 905, 1060	\ACRfullplfmt	258, 260, 261, 270, 271, 382, 384, 526, 527, 534, 535, 579, 581, 778, 779, 785, 787, 832, 834, 1032–1034, 1040, 1042, 1188–1190, 1196, 1197
\aa	27, 401, 651, 905, 1060	\Acrfullplfmt	258, 259, 261, 270, 271, 382, 384, 526, 527, 534, 535, 579, 581, 778, 779, 785, 787, 832, 834, 1032–1034, 1040, 1042, 1187, 1188, 1190, 1196, 1197
\Ac	274, 537, 788, 1043, 1199	\acrfullplfmt	258, 259, 261, 270, 271, 382, 384, 526, 527, 533, 535, 579, 580, 778, 779, 785, 787, 832, 833, 1032–1034, 1040, 1042, 1187, 1188, 1190, 1196, 1197
\ac	274, 537, 788, 1043, 1199	\acrlinkfootnote	275, 275, 538, 790, 1043, 1199
access (key)	352	\acrlinkfullformat	257, 257, 258, 525, 526, 777, 778, 1031, 1032, 1187, 1188
\accsuppglossaryentryfield	377, 553, 574, 806, 827	\ACRlong	167, 479, 731, 985, 1140
\accsuppglossarysubentryfield	377, 554, 574, 807, 827	\Acrlong	166, 166, 274, 479, 537, 730, 788, 984, 1043, 1140, 1199
\Acf	274, 537, 788, 1043, 1199	\acrlong	165, 273, 479, 536, 730, 788, 984, 1043, 1140, 1199
\acf	274, 537, 788, 1043, 1199	\ACRlongpl	169, 480, 732, 986, 1142
\Acfp	274, 537, 788, 1043, 1199	\Acrlongpl	168, 168, 274, 480, 537, 732, 788, 986, 1043, 1141, 1199
\acfp	274, 537, 788, 1043, 1199	\acrlongpl	167, 274, 480, 537, 731, 788, 985, 1043, 1141, 1199
\Acl	274, 537, 788, 1043, 1199	\acrnameformat	260, 526, 542, 584, 778, 795, 837, 1033, 1189
\acl	273, 536, 788, 1043, 1199	\acrnolinkfootnote	275, 538, 790, 1044, 1199
\aclp	274, 537, 788, 1043, 1199	acronym (option)	20
\aclp	274, 537, 788, 1043, 1199	\acronymentry	261, 262, 264–268, 270–273, 378–380, 383–386, 527–532, 534–536, 575–577, 579–582, 779, 781–784, 786–788, 828–830, 832–835, 1033–1043, 1189–1199
\Acp	274, 537, 788, 1043, 1199	\acronymfont	120, 121, 162–165, 172, 179, 260, 260, 262, 264–273, 366, 367, 369–371, 378–386, 455, 477, 478, 482, 485, 526, 528–536, 538–540, 542–547, 567, 569, 570, 575–583, 585, 586,
\acp	274, 537, 788, 1043, 1199		
\acrfootnote	275, 538, 544, 790, 791, 796, 1043, 1199		
\ACRfull	258, 525, 777, 1032, 1188		
\Acrfull	258, 274, 525, 537, 777, 788, 1032, 1043, 1188, 1199		
\acrfull	258, 258, 274, 525, 537, 777, 788, 1032, 1043, 1188, 1199		
\ACRfullfmt	257, 259, 261, 269, 271, 382, 384, 526, 527, 533, 535, 578, 580, 777, 779, 785, 787, 831, 833, 1032, 1034, 1040, 1041, 1187–1189, 1196, 1197		
\Acrfullfmt	257, 258, 261, 269, 271, 382, 384, 525, 527, 533, 535, 578, 580, 777, 779, 785, 787, 831, 833, 1032, 1034, 1040, 1041, 1187–1189, 1196, 1197		
\acrfullfmt	257, 258, 261, 269, 271, 382, 384, 525, 527, 533, 535, 578, 580, 777, 779, 785, 787, 831, 833, 1032, 1034, 1040, 1041, 1187–1189, 1196, 1197		
\acrfullformat	179, 257, 257, 485, 525, 537, 550, 736, 777, 789, 803, 991, 1031, 1032, 1146, 1187		
\ACRfullpl	259, 526, 778, 1033, 1188		
\Acrfullpl	259, 274, 526, 537, 778, 788, 1033, 1043, 1188, 1199		
\acrfullpl	259, 259, 274, 526, 537, 777, 788, 1032, 1033, 1043, 1188, 1199		

706, 728–730, 733, 736, 778, 779, 781–788, 790–792, 794, 796–800, 820, 822, 823, 828– 835, 837–839, 960, 982–984, 988, 991, 1033–1043, 1115, 1137–1139, 1143, 1146, 1189–1199	long-sc-short 265
acronymlists (option) 22	long-sc-short-desc 267, 379
\acronymname . 20, 48, 50, 397, 398, 411, 413, 647, 661, 663, 901, 902, 917, 919, 1056, 1072, 1073	long-short 263, 377
acronyms (option) 20	long-short-desc 266, 379
\acronymsort . . 261, 262, 264–268, 270–273, 378–380, 383, 384, 386, 527–532, 534–536, 575– 577, 579, 580, 582, 779, 781– 784, 786, 788, 828–830, 832, 833, 835, 1033–1043, 1189–1199	long-sm-short 265
\acronymtype 19, 20, 260–262, 397, 398, 527, 528, 537–550, 582– 586, 647, 778–780, 789–793, 795, 797–803, 836–839, 901, 902, 1033, 1034, 1056, 1189, 1190	long-sm-short-desc 267, 379
\acrpluralsuffix 256, 261, 264–266, 270–272, 378, 379, 383–385, 525, 527, 529–531, 534–540, 542–548, 550, 575, 576, 579– 581, 583–586, 777, 779, 781– 783, 786, 787, 789–793, 795– 801, 803, 828, 829, 832–839, 1031, 1034–1037, 1041, 1042, 1187, 1189, 1191–1193, 1196–1198	long-sp-short 264
\ACRshort . . . 163, 477, 729, 982, 1138	long-sp-short-desc 266
\Acrshort 162, 162, 273, 477, 536, 728, 788, 982, 1043, 1138, 1199	sc-short-long 266
\acrshort 161, 273, 476, 536, 728, 788, 982, 1043, 1137, 1199	sc-short-long-desc 268, 380
\ACRshortpl . . 165, 478, 730, 984, 1139	short-long 265, 378
\Acrshortpl 164, 164, 273, 478, 536, 729, 788, 983, 1043, 1139, 1199	short-long-desc 267, 380
\acrshortpl . . . 163, 273, 477, 536, 729, 788, 983, 1043, 1138, 1199	sm-short-long 266
acrstyles:	sm-short-long-desc 268, 380
dua 268, 381	\Acs 273, 536, 788, 1043, 1199
dua-desc 270, 383	\acs 273, 536, 788, 1043, 1199
footnote 270, 383	\Acsp 273, 536, 788, 1043, 1199
footnote-desc 272, 385	\acsp 273, 536, 788, 1043, 1199
footnote-sc 272, 385	\add@accent@ . . 27, 401, 650, 905, 1059
footnote-sc-desc 272, 386	\addcontentsline 58, 416, 666, 922, 1077
footnote-sm 272, 385	\addglossarytocaptions 49, 50, 412, 662, 917, 918, 1072
footnote-sm-desc 273, 386	\addtolength 339, 340, 600, 601, 638, 854, 892
	\advance 17, 18, 101, 107, 129, 306, 396, 442, 460, 619, 645, 646, 692, 711, 873, 900, 948, 953, 964, 1054, 1055, 1103, 1107, 1119
	\AE 27, 401, 651, 905, 1060
	\ae 27, 401, 651, 905, 1060
	altlist (style) 293
	altlistgroup (style) 294
	altlisthypergroup (style) 294
	altlong4col (style) 301
	altlong4col-booktabs (style) 304
	altlong4colborder (style) 301
	altlong4colheader (style) 301
	altlong4colheaderborder (style) 301
	altlongragged4col (style) 310
	altlongragged4col-booktabs (style) 305
	altlongragged4colborder (style) 311
	altlongragged4colheader (style) 311
	altlongragged4colheaderborder (style) 312
	\altnewglossary 77, 427, 677, 933, 1088
	altsuper4col (style) 324
	altsuper4colborder (style) 324
	altsuper4colheader (style) 324

altsuper4colheaderborder (style)	325	912 , 1010 , 1012 , 1030 , 1048 ,
altsuperragged4col (style)	329	1050 , 1066 , 1165 , 1167 , 1172 , 1186
altsuperragged4colborder (style)	331	<code>\bfseries</code> . 297–300, 302, 303, 308,
altsuperragged4colheader (style)	330	310–312, 320–325, 327, 329,
altsuperragged4colheaderborder		331, 615–618, 620–623, 627–
(style)	331	634, 868–871, 874–876, 881–887
almtree (style)	338	<code>\bgroup</code> 26, 98, 136, 179,
almtreegroup (style)	340	213, 228, 253, 401, 440, 485,
almtreehypergroup (style)	340	509, 511, 650, 691, 717, 737,
<code>\andname</code>	48 , 216 , 412 , 507 ,	760, 762, 905, 947, 970, 991,
	662 , 759 , 917 , 1015 , 1072 , 1170	1013, 1016, 1018, 1030, 1059,
<code>\AnyTrackedLanguages</code>		1101, 1125, 1147, 1169, 1173, 1186
	50, 341, 342, 412,	<code>\bool</code> 139, 225, 1127
	588, 604, 662, 841, 857, 918, 1073	<code>\boolean</code> 549, 802
<code>\appto</code>	22,	<code>\boolfalse</code> . . . 39, 408, 658, 913, 1068
	28, 83, 92, 93, 209, 343, 354,	<code>\booltrue</code> 408, 658
	399, 402, 430, 436, 555, 605,	<code>\bottomrule</code> . . . 302, 303, 617, 618, 871
	648, 651, 680, 686, 687, 808,	<code>\box</code> 306, 619, 873
	858, 903, 906, 936, 942, 943,	
	1011, 1057, 1060, 1091, 1097, 1167	
		C
<code>\AtBeginDocument</code>		<code>\c</code> 26, 27, 401, 651, 905, 1060
	20, 66, 89, 107, 181, 201,	<code>\c@equation</code> . . 129, 460, 711, 964, 1119
	397, 422, 434, 446, 486, 499,	<code>\c@glossaryentry</code>
	647, 672, 684, 696, 738, 751,	14, 394, 644, 898, 1053
	901, 928, 940, 952, 992, 1006,	<code>\c@glossarysubentry</code>
	1056, 1082, 1095, 1107, 1148, 1162	15, 395, 644, 898, 1053
<code>\AtEndDocument</code>		<code>\c@page</code> 207–209, 211, 212, 503–505,
	36, 38, 89, 114, 200, 204, 220,	755–757, 1010–1012, 1166–1168
	221, 287, 407, 408, 434, 449,	<code>\capitalisewords</code> 175, 734, 989, 1145
	499, 502, 509, 510, 610, 656,	<code>\catcode</code> 36,
	658, 684, 700, 751, 754, 761,	89, 136, 434, 657, 684, 717,
	762, 863, 911, 913, 940, 957,	912, 940, 970, 1066, 1095, 1125
	1006, 1009, 1017, 1047, 1066,	<code>\cGls</code> 115,
	1067, 1095, 1111, 1161, 1164, 1173	115 , 450, 700, 701, 957, 958, 1112
automake (option)	37	<code>\cgl</code> s 115 , 115 , 449, 450, 700, 957, 1112
automakegloss (option)	38	<code>\cGlsformat</code> 113, 116 , 449,
automakeglosslite (option)	38	450, 699, 701, 956, 958, 1111, 1112
		<code>\cgl</code> sformat 113, 115 , 448,
		450, 699, 700, 956, 957, 1111, 1112
	B	<code>\cGlspl</code>
<code>\b</code>	27, 401, 651, 905, 1060	116, 116 , 117, 450, 701, 958, 1113
<code>\begin</code>	187,	<code>\cgl</code> spl . 116, 116 , 450, 701, 958, 1113
	233, 236, 292, 296–301, 304,	<code>\cGlsplformat</code> 114, 117 , 449,
	305, 307–331, 489, 514, 592,	450, 700, 701, 957, 958, 1111, 1113
	612, 614–634, 741, 766, 845,	<code>\cgl</code> splformat 113, 116 , 448,
	866–887, 996, 1021, 1151, 1177	450, 699, 701, 956, 958, 1111, 1113
<code>\BeginAccSupp</code>	351, 554, 807	<code>\changes</code> 179
<code>\begin</code> group	6,	<code>\char</code> 249, 522, 774, 1028, 1184
	8, 36, 206, 211, 220, 253, 390,	<code>\cleardoublepage</code>
	391, 503, 504, 524, 640, 641,	57, 416, 666, 922, 1076, 1077
	657, 754, 756, 776, 894, 895,	

<code>\clearpage</code>	...	57, 416, 666, 922, 1076	
<code>\closeout</code>	36, 89, 188, 190, 195, 205,	407, 434, 490, 492, 495, 502,	
	657, 684, 742, 743, 747, 754,	912, 940, 997, 998, 1002, 1009,	
	1066, 1095, 1152, 1154, 1157, 1165		
<code>\compatglossarystyle</code>	594–604, 847–857	
<code>compatible-2.07</code> (option)	39	
<code>compatible-3.07</code> (option)	39	
<code>\compatibleglossentry</code>	246,	
	351, 520, 521, 553, 772, 773, 806		
<code>\compatiblesubglossentry</code>	247, 351, 521, 554, 772, 773, 807	
<code>\copy</code>	306, 619, 873	
<code>\count@</code>	237, 515, 767, 1022, 1178	
<code>counter</code> (key)	80	
<code>counter</code> (option)	22	
counters:			
<code>glossaryentry</code>	14	
<code>glossarysubentry</code>	15	
<code>counterwithin</code> (option)	14	
<code>\cs</code>	26, 43–45, 222, 224–227,	
	232, 233, 236, 240, 241, 286, 287		
<code>\csdef</code>	24, 92–94, 104, 105, 112, 113,	227, 228, 251, 263, 400, 436,	
	437, 444, 448, 510, 511, 523,	528, 594, 649, 686–688, 695,	
	698, 699, 762, 763, 775, 780,	847, 904, 942–944, 951, 955,	
	956, 1017, 1018, 1029, 1035,	1058, 1097–1099, 1105, 1106,	
	1110, 1111, 1173, 1174, 1185, 1190		
<code>\csedef</code>	114, 208, 449, 504,	
	700, 755, 957, 1011, 1112, 1166		
<code>\csgdef</code>	19, 54, 75, 78,	
	113, 114, 216, 240, 414, 426,	428, 448, 449, 507, 517, 664,	
	676, 678, 699, 700, 759, 768,	901, 920, 932, 934, 956, 957,	
	1015, 1024, 1055, 1075, 1087,	1089, 1111, 1112, 1170, 1179, 1180	
<code>\cslet</code>	82, 83, 98,	
	105, 232, 430, 440, 444, 514,	680, 691, 695, 765, 936, 947,	
	951, 1021, 1091, 1101, 1106, 1176		
<code>\csname</code>	15–18, 27,	
	42, 48, 50, 56, 57, 60, 62, 63,	66, 67, 71, 76–78, 85, 91–97,	
	100–107, 110, 111, 127, 129,	130, 141–146, 162–169, 180,	
	181, 185–187, 191–195, 201,	204–206, 208, 213, 214, 218–	
	220, 228, 250, 251, 255, 279–	284, 338, 339, 344, 355, 356,	
	359, 363, 369–371, 387–389,	395–397, 401, 410–413, 415,	
	416, 418–420, 422–424, 427,	428, 432, 435–439, 441–445,	
	447, 457, 459, 460, 468–470,	477–481, 485, 486, 488, 489,	
	493–495, 499, 502–511, 517,	520–523, 525, 551–554, 556,	
	557, 559, 561, 569, 570, 587,	588, 590, 600, 601, 605, 606,	
	610, 637, 638, 645, 646, 651,	660–663, 665, 666, 668–670,	
	672–674, 677, 678, 682, 686–	690, 692–696, 698, 707, 708,	
	710, 711, 719–722, 728–732,	737, 738, 740, 741, 745–747,	
	751, 753–758, 760, 761, 763,	769, 772–776, 789, 790, 804–	
	807, 809, 810, 812, 814, 822,	823, 840–843, 854, 858, 859,	
	863, 891, 892, 899, 900, 905,	915, 917–919, 921, 923, 925–	
	928, 930, 933, 934, 938, 942–	946, 948–955, 963–965, 973–	
	975, 982–986, 991, 992, 994,	995, 999–1002, 1006, 1009–	
	1011, 1013, 1014, 1016–1018,	1029, 1031, 1044–1046, 1053–	
	1055, 1059, 1060, 1070, 1072,	1073, 1076, 1078, 1080, 1082,	
	1083, 1085, 1088, 1092, 1093,	1097–1101, 1103–1107, 1109,	
	1110, 1118–1120, 1129–1131,	1138–1142, 1147, 1148, 1150,	
	1151, 1155–1157, 1162, 1164–	1166, 1168, 1169, 1171, 1172,	
	1174, 1184, 1185, 1187, 1200–1202		
<code>\csshow</code>	283, 552, 805, 1046, 1201	
<code>\csuse</code>	50, 53,	
	54, 75, 82, 85, 86, 93, 94, 170,	195–197, 228, 236, 238, 240,	
	242, 263, 344, 413, 414, 426,	432, 433, 436, 437, 457, 496,	
	497, 511, 513–516, 518, 523,	528, 595–605, 663, 664, 676,	

682, 683, 687, 688, 708, 747–749, 763, 765–770, 774, 780, 848–857, 859, 918, 920, 932, 936, 938, 943, 944, 986, 1002–1004, 1018, 1020–1022, 1024, 1025, 1035, 1073, 1075, 1087, 1091–1093, 1098, 1099, 1142, 1158, 1159, 1174, 1176–1181, 1191

`\csxdef` . . . 103, 114, 443, 449, 694, 700, 789, 790, 950, 957, 1105, 1112

`\currentglossary` 14, 15, 54, 219, 285, 287, 288, 394, 395, 414, 509, 644, 664, 761, 898, 899, 920, 1016, 1046, 1047, 1053, 1075, 1172

`\currentglosssubentry` 15, 244, 395, 519, 644, 771, 899, 1026, 1053, 1182

`\CurrentOption` . 42, 343, 350, 410, 553, 605, 660, 806, 858, 915, 1070

`\CurrentTrackedLanguage` 50, 342, 413, 588, 605, 663, 841, 858, 918, 1073

`\CurrentTrackedTag` 50, 341, 342, 413, 588, 604, 663, 841, 858, 918, 1073

`\CustomAcronymFields` . . 278, 550, 803

`\CustomNewAcronymDef` 278, 550, 803, 804

D

`\d` 27, 401, 651, 905, 1060

`\datatool` 26, 222, 225

`\datatoolasciistart` 223

`\datatoolctrlboundary` 223, 225

`\day` 185, 189, 488, 491, 590, 593, 739, 742, 843, 846, 994, 997, 1150, 1152

`debug` (option) 7

`\DeclareAcronymList` 20, 21, 22, 261, 262, 397–399, 527, 528, 538, 540, 541, 543, 545, 547, 549, 550, 647, 648, 778, 780, 790, 792, 794, 796, 798, 800, 802, 803, 901, 902, 1033, 1034, 1056, 1057, 1189, 1190

`\DeclareCurrentRelease` 4, 285, 289, 291, 295, 302, 306, 312, 318, 325, 332, 341–343, 350, 389, 639, 893

`\DeclareListParser` 199, 498, 750, 1005, 1161

`\DeclareOption` . 11, 343, 350, 393, 553, 605, 643, 806, 858, 897, 1052

`\DeclareOptionX` 11, 393, 643, 897, 1052

`\DeclareRelease` . 4, 285, 289, 291, 295, 302, 306, 312, 318, 325, 332, 341–343, 350, 389, 639, 893

`\DeclareRobustCommand` 51, 413, 545, 663, 798, 919, 1074

`\def` 11, 12, 14–21, 25–27, 33, 34, 36, 38, 40, 41, 46–48, 50–52, 55, 58, 61–67, 71, 75–77, 79–84, 86, 96, 98–103, 105, 108, 109, 113–117, 123, 124, 127–138, 141–169, 180, 181, 184, 189–193, 195–198, 200–202, 205, 208, 209, 211–215, 218, 227, 231, 236, 237, 239, 241, 243, 249–255, 258–260, 288–290, 317, 318, 338–341, 343, 346–349, 352–355, 369–376, 393–398, 400–402, 406, 408–411, 413, 414, 417, 419–422, 424, 427–431, 433, 438, 440–444, 446, 448–450, 457–481, 485, 486, 488, 491–500, 502, 504–510, 513–518, 522–527, 537–539, 541–550, 553–556, 569–573, 583–587, 589, 590, 593–595, 600, 601, 605–612, 625, 626, 637–639, 643–648, 650, 651, 655, 657–661, 663, 664, 667, 669–672, 674, 677–681, 683, 689, 691–695, 697, 699–701, 707, 708, 710–732, 737–739, 742, 744–749, 751, 752, 754–760, 762, 765–770, 773–778, 789–795, 797–803, 806–809, 822–827, 836–840, 842, 843, 846–848, 853, 854, 858, 860–865, 879, 880, 891, 892, 897–902, 904, 905, 910–914, 916, 917, 919, 920, 922, 923, 925–928, 930, 932–937, 939, 945, 947–951, 953, 956–958, 962–986, 991, 992, 994, 997–1004, 1006, 1007, 1009, 1011–1014, 1016, 1017, 1021–1025, 1028–1033, 1047, 1051–1057,

1059, 1060, 1064–1075, 1077,
1079–1085, 1087–1094, 1100–
1106, 1108, 1111–1113, 1116–
1127, 1129–1142, 1147–1149,
1152, 1154–1164, 1166–1173,
1176–1181, 1184–1186, 1188, 1189
\def@gl@x@dy@check@backslash
. 137, 138,
466, 467, 718, 971, 972, 1126, 1127
\DefaultNewAcronymDef 275,
387, 537, 538, 582, 789, 790, 835
\defgl@display 123, 457, 707
\defgl@display@first . . . 123, 457, 708
\defgl@entry@fmt 75,
77, 78, 262, 275, 426–428, 457,
528, 537, 538, 540, 541, 543,
545, 548, 550, 676–678, 707,
708, 780, 789, 790, 792, 794,
796, 798, 801, 803, 932–934,
1034, 1043, 1087–1089, 1190, 1199
\define@boolkey 10, 12–15, 20, 28,
32–34, 39, 124, 242, 392–395,
397, 402, 405, 406, 408, 458,
518, 642–645, 647, 651, 652,
654, 655, 658, 709, 770, 896–
899, 901, 906, 909, 910, 913,
962, 1025, 1051–1053, 1056,
1060, 1061, 1064, 1068, 1117, 1181
\define@choicekey 7, 9, 10, 15, 29,
30, 33, 37, 40, 42, 82, 241, 242,
390, 392, 393, 395, 403, 404,
406–408, 430, 517, 518, 640,
642, 645, 652, 653, 655, 657,
658, 680, 769, 770, 894, 896,
899, 907, 908, 910, 912, 914,
915, 936, 1024, 1025, 1049–
1051, 1053, 1061, 1062, 1064,
1067, 1068, 1070, 1090, 1180, 1181
\define@key 11,
14, 22, 23, 29, 32, 34, 79–84,
92, 123, 124, 181, 241, 243,
343, 352, 353, 393, 395, 399,
403, 406, 428–431, 436, 457,
458, 486, 517, 518, 554, 555,
605, 643, 644, 648, 653, 655,
678–681, 686, 687, 708, 737,
769, 770, 807, 808, 858, 897,
898, 902, 903, 907, 909, 910,
934–937, 942, 943, 961, 962,
992, 1024, 1025, 1051, 1053,
1057, 1062, 1063, 1065, 1089–
1092, 1097, 1116, 1148, 1180, 1181
\DefineAcronymSynonyms 46,
273, 275, 410, 536, 537, 660,
788, 789, 916, 1043, 1070, 1199
\delimN 54, 187, 199, 239, 253, 414,
490, 498, 516, 524, 592, 664,
741, 750, 768, 775, 776, 845,
920, 996, 1005, 1023, 1030,
1075, 1152, 1161, 1179, 1185, 1186
\delimR 54, 187,
252, 253, 414, 490, 523, 524,
592, 664, 741, 775, 845, 920,
996, 1029, 1030, 1075, 1152, 1185
description (key) 79
description (option) 32
descriptionaccess (key) 352
\DescriptionDUANewAcronymDef . .
. 276, 540, 541, 792, 794
\DescriptionFootnoteNewAcronymDef
. 276,
387, 539, 540, 583, 791, 792, 836
\descriptionname . . . 48, 50, 297–
300, 302, 303, 308, 310–312,
320–325, 327, 329, 331, 411,
413, 615–618, 620–623, 627–
634, 661, 663, 868–871, 874–
876, 881–887, 917, 919, 1072, 1073
\DescriptionNewAcronymDef . 276,
387, 542, 543, 584, 794, 796, 837
descriptionplural (key) 79
descriptionpluralaccess (key) . . 352
\DH 27, 402, 651, 905, 1060
\dh 27, 402, 651, 905, 1060
\dimen@ . . 264, 306, 337, 530, 619,
637, 781, 873, 890, 891, 1036, 1192
\directlua 43, 915, 1070
\disable@keys . 45, 410, 660, 916, 1070
\disablemakegloss (option) 35
\do 31, 42, 47, 49, 59,
60, 67, 89, 91, 131, 180, 185,
187, 197–199, 207, 215, 227,
228, 262, 405, 410, 411, 417,
418, 423, 434, 435, 461, 485,
488, 489, 497–499, 503, 507,
510, 511, 528, 538, 540, 541,
543, 545, 547, 549, 551, 590,
610, 654, 660, 661, 667, 668,
673, 685, 686, 712, 737, 739–
741, 749, 750, 755, 759, 762,

780, 790, 792, 794, 796, 798, 800, 802, 804, 843, 863, 909, 915, 917, 923, 924, 928, 941, 942, 966, 992, 994, 995, 1004, 1005, 1010, 1014, 1017, 1018, 1034, 1046, 1047, 1063, 1070– 1072, 1077, 1078, 1083, 1095, 1096, 1120, 1147, 1150, 1151, 1160, 1161, 1166, 1170, 1173, 1190	<code>\DTLsubstituteall</code>
<code>\do@glo@storeentry</code> 131, 461, 712, 966, 1120
<code>\do@gls@link@checkfirsthyper</code>	<code>\dtlwordindexcompare</code>
. 126, 128, 141–146, 162–169, 369, 370, 458, 459, 468–471, 476–481, 569, 570, 709, 710, 719–722, 728–732, 822, 823, 962, 963, 973–976, 982–986, 1117, 1118, 1129–1131, 1137–1142 229, 511, 763, 1019, 1174
<code>\do@gls@xdycheckbackslash</code>	<code>dua</code> (acrstyle) <u>268</u> , <u>381</u>
. 130, 131, 461, 712, 965, 1120	<code>dua</code> (option) <u>33</u>
<code>\do@gls@disablehyperinlist</code>	<code>dua-desc</code> (acrstyle) <u>270</u> , <u>383</u>
. <u>127</u> , 128, 459, 710, 963, 1118	<code>\DUANewAcronymDef</code>
<code>\do@gls@haschildren</code> <u>277</u> , 548, 549, 801, 802
. 71, 424, 674, 930, 1084, 1085	
<code>\do@write@gls@labelsname</code>	
. 36, <u>36</u> , 37, 657, 911, 912, 1066	
<code>\doifglossarynoexistsordo</code>	
. <u>70</u> , 76, 424, 427, 674, 677, 930, 933, 1084, 1087	
<code>\dtl@ifsingle</code> 521, 773	
<code>\dtl@insertinto</code>	
. 229, 511, 763, 1019, 1174	
<code>\dtl@sortresult</code> 229, 230, 511, 512, 763, 764, 1019, 1174, 1175	
<code>\DTLassignlettergroup</code> 226	
<code>\dtlcompare</code> 230, 512, 764, 1019, 1175	
<code>\dtlicompare</code> 230, 512, 764, 1019, 1175	
<code>\DTLifinlist</code> 79, 127, 428, 459, 678, 710, 934, 963, 1089, 1118	
<code>\DTLifint</code> 249, 522, 774, 1028, 1184	
<code>\dtlletterindexcompare</code>	
. 229, 512, 763, 1019, 1175	
<code>\dtlnonlettergroup</code> 225	
<code>\dtlnumbergroup</code> 225	
<code>\DTLsortedletter</code> 226	
<code>\DTLsortletterhandler</code> 224	
<code>\DTLsortwordcasehandler</code> 225	
<code>\DTLsortwordhandler</code> 224, 225	
	E
	<code>\eappto</code> 428, 444, 504
	<code>\edef</code> 47, 50, 58–65, 76, 85, 86, 89, 91, 129–138, 183, 189– 193, 199, 205, 220, 221, 229, 230, 237, 249, 253, 255, 396, 398, 411, 412, 417–424, 427, 428, 432, 434, 435, 438–441, 444, 445, 457, 459–467, 485– 487, 490, 492–499, 502, 504– 507, 509–512, 515, 519, 522, 524, 525, 537, 539, 540, 542, 544, 546, 548, 582–586, 588– 590, 593, 609, 661, 662, 667– 672, 677, 682, 684, 686, 707, 708, 710–718, 738, 742, 744– 750, 754, 757, 758, 761–764, 767, 774, 776, 789, 791, 792, 794, 797, 799, 801, 835–839, 841, 842, 844, 846, 917, 918, 922–927, 933, 938, 940, 942, 964–972, 993, 997–1005, 1009, 1017, 1019, 1022, 1028, 1030, 1031, 1071, 1073, 1077–1082, 1088, 1092, 1093, 1095, 1096, 1119–1127, 1148, 1152, 1154– 1156, 1158, 1159, 1161, 1165, 1172–1175, 1178, 1184, 1186, 1187
	<code>\egroup</code> 26, 98, 180, 213, 228, 253, 401, 440, 486, 509, 511, 650, 691, 737, 761, 763, 905, 947, 992, 1013, 1017, 1018, 1030, 1059, 1101, 1148, 1169, 1174, 1186
	<code>\else</code> 8, 13–15, 18– 21, 23, 25, 28, 29, 36, 38, 39, 42, 46, 47, 51–53, 55–62, 64– 67, 75, 82, 84, 101, 102, 105– 108, 113, 114, 124, 127–138, 172, 183–185, 188, 190, 191, 193–198, 205–209, 211–215,

219, 237, 243–245, 250, 252,
 253, 264, 277, 278, 293, 296,
 298, 300, 302–304, 306, 308,
 309, 311, 319, 321, 323, 327,
 328, 330, 333–335, 337–340,
 360, 391, 394–400, 402, 403,
 407, 408, 410, 411, 413–423,
 430, 431, 441, 442, 445, 446,
 448, 449, 459–471, 482, 487,
 488, 490, 492–497, 502–507,
 509, 515, 518, 519, 522–524,
 530, 540, 541, 543, 545, 547–
 549, 551, 559, 589–594, 598–
 601, 610, 612, 614–622, 627–
 629, 631–633, 635–639, 641,
 644, 646–652, 657, 658, 660,
 661, 663–673, 680, 681, 692,
 693, 695, 696, 699, 700, 709–
 718, 720–722, 733, 738, 739,
 741–744, 746, 747, 749, 754–
 760, 767, 770, 771, 774–776,
 781, 792, 793, 795–798, 800,
 802, 804, 812, 843–848, 851–
 854, 863, 866, 868–876, 881,
 882, 885–892, 895, 898–904,
 906, 907, 911–913, 915–917,
 919, 921–928, 932, 936, 937,
 948, 949, 951–953, 956, 957,
 962–972, 987, 988, 993, 994,
 996–1004, 1009–1014, 1016,
 1022, 1025–1030, 1036, 1044,
 1047, 1049, 1050, 1052, 1053,
 1055–1061, 1066–1068, 1070,
 1071, 1074–1083, 1087, 1091,
 1092, 1103, 1104, 1106–1108,
 1111, 1117–1127, 1143, 1149,
 1150, 1152–1154, 1156–1160,
 1165–1170, 1172, 1178, 1181,
 1182, 1184–1186, 1192, 1199, 1200
 \emph 215, 254, 507, 524,
 759, 776, 1014, 1031, 1170, 1186
 \empty 253,
 350, 524, 553, 776, 806, 1030, 1186
 \encodingdefault
 27, 401, 651, 905, 1059
 \end 187, 233, 236, 292, 296–301, 304,
 305, 307–331, 489, 514, 592,
 612, 614–634, 741, 766, 845,
 866–887, 996, 1021, 1151, 1177
 \end@doifinlist
 . 58, 59, 417, 667, 922, 923, 1077
 \end@getprefix
 214, 506, 758, 1013, 1169
 \end@glis@islistofacronyms
 21, 398, 648, 902, 1057
 \EndAccSupp 351, 554, 807
 \endcsname 15–
 18, 27, 42, 48, 50, 56, 57, 60,
 62, 63, 66, 67, 71, 76–78, 85,
 91–97, 100–107, 110, 111, 127,
 129, 130, 141–146, 162–169,
 180, 181, 185–187, 191–195,
 201, 204–206, 208, 213, 214,
 218–220, 228, 250, 251, 255,
 279–284, 338, 339, 344, 355,
 356, 359, 363, 369–371, 387–
 389, 395–397, 401, 410–413,
 415, 416, 418–420, 422–424,
 427, 428, 432, 435–439, 441–
 445, 447, 457, 459, 460, 468–
 470, 477–481, 485, 486, 488,
 489, 493–495, 499, 502–511,
 517, 520–523, 525, 551–554,
 556, 557, 559, 561, 569, 570,
 587, 588, 590, 600, 601, 605,
 606, 610, 637, 638, 645, 646,
 651, 660–663, 665, 666, 668–
 670, 672–674, 677, 678, 682,
 686–690, 692–696, 698, 707,
 708, 710, 711, 719–722, 728–
 732, 737, 738, 740, 741, 745–
 747, 751, 753–758, 760–763,
 769, 772–776, 789, 790, 804–
 807, 809, 810, 812, 814, 822,
 823, 840–843, 854, 858, 859,
 863, 891, 892, 899, 900, 905,
 915, 917–919, 921, 923, 925–
 928, 930, 933, 934, 938, 942–
 946, 948–955, 963–965, 973–
 975, 982–986, 991, 992, 994,
 995, 999–1002, 1006, 1009–
 1011, 1013, 1014, 1016–1018,
 1029, 1031, 1044–1046, 1053–
 1055, 1059, 1060, 1070, 1072,
 1073, 1076, 1078, 1080, 1082,
 1083, 1085, 1088, 1092, 1093,
 1097–1101, 1103–1107, 1109,
 1110, 1118–1120, 1129–1131,
 1138–1142, 1147, 1148, 1150,
 1151, 1155–1157, 1162, 1164–

1166, 1168, 1169, 1171, 1172,
 1174, 1184, 1185, 1187, 1200–1202
`\endfoot` 297–300,
 302, 303, 308, 310, 312, 614–
 618, 620–623, 868–871, 874–876
`\endgroup` 6,
 8, 36, 207, 212, 220, 253, 390,
 392, 503, 505, 524, 640, 641,
 657, 755, 757, 776, 894, 895,
 912, 1010, 1012, 1030, 1048,
 1050, 1066, 1166, 1168, 1172, 1186
`\endhead` 297–300,
 302, 303, 308, 310–312, 614–
 618, 620–623, 868–871, 874–876
`\endtheglossary` 6, 390, 640, 894, 1048
`entrycounter` (option) [14](#)
`\entryname` [48](#), 50, 297–
 300, 302, 303, 308, 310–312,
 320–325, 327, 329, 331, 411,
 413, 615–618, 620–623, 627–
 634, 661, 663, 868–871, 874–
 876, 881–887, 917, 919, 1072, 1073
 environments:
 `theglossary` [245](#)
`\epreto` 107, 953, 1107
`\equal` 29, 46, 57, 129, 198, 403,
 411, 416, 460, 498, 522, 610,
 653, 661, 666, 710, 750, 774,
 863, 907, 916, 922, 964, 1005,
 1046, 1062, 1071, 1076, 1119, 1160
`esclocations` (option) [12](#)
`\exp` 26, 45, 222–226, 234–236, 288
`\expandafter` 8, 16–18, 25–27, 42,
 47, 50, 60, 62, 63, 65–68, 71,
 76, 77, 79, 85, 86, 89–97, 100–
 104, 106, 107, 109–111, 122,
 123, 127, 129–137, 170–172,
 175, 180, 183, 187, 191, 192,
 194, 205–207, 209, 211–213,
 215, 219, 220, 227–229, 237,
 246, 251, 253, 255, 279–284,
 338, 344, 355, 356, 359, 360,
 387–389, 392, 395–397, 400,
 401, 410–412, 418–424, 427,
 428, 432, 434–439, 441–447,
 457, 459–466, 482, 486, 487,
 489, 492–495, 502–505, 507,
 509, 511, 515, 520, 521, 523–
 525, 538, 544, 551–554, 556,
 557, 559, 582, 587–590, 592,
 593, 605, 610, 637, 641, 645,
 646, 650, 651, 660–662, 668–
 674, 677, 678, 682, 685–690,
 692–698, 707, 708, 710–718,
 733, 735, 737, 738, 741, 744–
 747, 754–757, 759, 760, 763,
 766, 767, 772, 773, 775, 776,
 789–791, 796, 804–807, 809,
 810, 812, 835, 840–843, 845,
 846, 858, 863, 891, 895, 899,
 900, 904, 905, 915, 917, 918,
 923, 925–930, 933, 934, 938,
 941–946, 948–955, 961, 963–
 971, 987–989, 992, 993, 995,
 998–1001, 1009–1014, 1016,
 1018, 1022, 1023, 1027, 1029–
 1031, 1044–1047, 1050, 1054,
 1055, 1059, 1060, 1070, 1071,
 1073, 1078, 1080, 1082–1084,
 1088, 1089, 1092, 1093, 1095–
 1101, 1103–1110, 1116, 1118–
 1126, 1142, 1143, 1145, 1147,
 1148, 1151, 1154–1157, 1165–
 1168, 1170, 1172, 1174, 1177,
 1178, 1183, 1185–1187, 1200–1202
`\expandonce` 85, 86, 96,
 131, 132, 191–193, 208, 229,
 230, 261, 432, 433, 461, 493,
 494, 504, 512, 520, 521, 527,
 537, 539, 540, 542, 544, 546,
 548, 553, 554, 682, 683, 690,
 712, 713, 745, 746, 755, 756,
 763, 764, 772, 773, 779, 789–
 791, 793, 795, 797, 799, 801,
 806, 807, 938, 939, 946, 966,
 999–1001, 1011, 1019, 1033,
 1092, 1093, 1100, 1120, 1121,
 1155, 1156, 1166, 1174, 1175, 1189
`\ExplSyntaxOff` 6, 26, 45, 108, 139,
 227, 236, 241, 249, 250, 288,
 893, 1028, 1048, 1108, 1128, 1184
`\ExplSyntaxOn` 4, 26, 43, 108, 139,
 221, 232, 240, 249, 250, 285,
 893, 1028, 1048, 1108, 1127, 1184

F
`\fi` 7, 8, 10, 11,
 13–16, 18–21, 23, 25, 28, 29,
 31–33, 35, 36, 38, 39, 41, 42,
 45–47, 51–53, 55–67, 75, 77,

82, 85, 96, 100–103, 105–108,
113, 114, 124, 127–138, 172,
181, 183–185, 188, 190, 191,
193, 194, 196–199, 205–209,
211–216, 218–221, 237, 242–
245, 250, 252–254, 264, 275,
277, 278, 293, 296, 298, 300,
302–306, 308, 309, 311, 319,
321, 323, 327, 328, 330, 334–
340, 350, 360, 390–400, 402–
404, 407–411, 413–423, 428,
430, 432, 441–443, 445, 446,
448, 449, 459–471, 482, 486–
488, 490, 492–499, 502–510,
515, 518–520, 522–524, 530,
537, 538, 540, 541, 543, 545,
547–551, 553, 559, 589–592,
594, 595, 598–601, 610, 612,
614–622, 627–629, 631–633,
635–650, 652, 654, 657–661,
663–673, 678, 680, 682, 690,
692–696, 699, 700, 709–718,
720–722, 733, 738, 739, 741–
744, 746–750, 754–762, 767,
769–771, 774–776, 781, 789,
790, 792–798, 800–804, 806,
812, 842–845, 847, 848, 851–
854, 863, 866, 868–876, 881,
882, 885–892, 894–904, 906–
917, 919–928, 932, 934, 936,
938, 946, 948–953, 956, 957,
962–972, 988, 992–994, 996,
998–1001, 1003–1005, 1009–
1017, 1022, 1025–1027, 1029–
1031, 1036, 1043, 1044, 1047–
1059, 1061, 1063–1071, 1074–
1083, 1087, 1088, 1091, 1092,
1100, 1103–1108, 1111, 1117–
1127, 1143, 1148–1150, 1152–
1161, 1165–1173, 1178, 1180–
1182, 1184–1186, 1192, 1199, 1200

`\findrootlanguage` 65, 422, 672
`first` (key) 80
`firstaccess` (key) 352
`\firstacronymfont` 122, 260,
263–265, 271, 368, 377–379,
383, 384, 456, 526, 529, 530,
534, 535, 538, 542, 544, 546,
568, 569, 575, 576, 580, 707,
778, 780–782, 786, 790, 794,
796, 799, 821, 822, 828, 829,
833, 961, 1033, 1035–1037,
1041, 1116, 1189, 1191, 1192, 1197

`firstplural` (key) 80
`firstpluralaccess` (key) 352
`\fmtversion` 27, 402, 651, 905, 1060
`\footnote` 271, 275, 384,
534, 535, 538, 580, 786, 790,
833, 1041, 1043, 1044, 1197, 1199
`footnote` (acrstyle) 270, 383
`footnote` (option) 32
`footnote-desc` (acrstyle) 272, 385
`footnote-sc` (acrstyle) 272, 385
`footnote-sc-desc` (acrstyle) 272, 386
`footnote-sm` (acrstyle) 272, 385
`footnote-sm-desc` (acrstyle) 273, 386
`\FootnoteNewAcronymDef` 276,
387, 544, 545, 585, 797, 798, 838
`\footnotesize` 8, 391, 641, 895, 1050
`\forallacronyms` 67, 423, 673, 928, 1083
`\forallglossaries` 67, 68, 204,
218, 337, 423, 502, 508, 637,
673, 754, 760, 891, 928, 929,
1009, 1015, 1016, 1083, 1164, 1171
`\forallglentries` 36,
68, 111, 114, 182, 407, 423,
447, 449, 486, 657, 673, 698,
700, 738, 911, 929, 955, 957,
993, 1066, 1083, 1110, 1112, 1148
`\ForEachTrackedDialect`
. 50, 341, 342, 412,
588, 604, 662, 841, 857, 918, 1073
`\forallglentries`
. . . . 67, 68, 71, 105, 231, 337,
423, 424, 444, 513, 637, 673,
674, 695, 765, 891, 928–930,
951, 1021, 1083, 1084, 1106, 1176
`\foralllistcsloop`
222, 227, 233, 236, 510, 514,
762, 766, 1017, 1021, 1173, 1177
`\foralllistloop` 202, 203,
239, 500, 501, 516, 752, 753,
768, 1007, 1008, 1023, 1163, 1179

G

`\gdef` 17, 36, 60, 76, 95, 101, 102,
206, 243, 287, 396, 418, 427,
438, 441, 442, 503, 518, 610,
646, 657, 668, 677, 689, 692,
693, 754, 770, 863, 900, 912,

923, 933, 945, 948, 949, 1010, 1025, 1047, 1055, 1066, 1078, 1088, 1100, 1103, 1104, 1165, 1181	<code>\glo@value</code> 90, 434, 685, 941, 1095, 1096
<code>\Genacrfullformat</code> 122, <u>122</u> , 261, 263–265, 271, 368, <u>368</u> , 378, 384, 456, 527, 529, 530, 534, 568, 575, 580, 707, 779–782, 786, 821, 828, 833, 961, 1034– 1036, 1041, 1116, 1189–1192, 1197	<code>\global</code> 17, 18, 85, 88, 98, 103, 104, 110, 111, 206, 220, 238, 243, 306, 396, 397, 432, 433, 440, 444, 447, 503, 509, 515, 516, 518, 619, 645, 646, 682, 684, 691, 694, 698, 755, 761, 767, 770, 873, 899, 900, 938, 939, 947, 950, 955, 1010, 1017, 1022, 1023, 1026, 1054, 1055, 1092, 1094, 1101, 1105, 1109, 1110, 1165, 1172, 1178, 1181
<code>\genacrfullformat</code> 121, 122, <u>122</u> , 261, 263– 265, 271, 368, <u>368</u> , 377, 378, 383, 456, 527, 529, 530, 534, 568, 575, 580, 707, 779–782, 786, 821, 828, 833, 961, 1034– 1036, 1041, 1116, 1189–1192, 1197	<code>\glolinkprefix</code> 128, <u>128</u> , 180, 246, 459, 486, 520, 710, 737, 772, 964, 992, 1027, 1118, 1119, 1148, 1183
<code>\GenericAcronymFields</code> 261, <u>262</u> , 263–270, 272, 273, 377– 380, 382, 383, 386, 527–534, 536, 575–580, 582, 779–781, 783–786, 788, 828–833, 835, 1034–1036, 1038–1043, 1189–1199	<code>\glosortentrieswarning</code> 23, <u>23</u> , 227, 399, 510, 649, 762, 903, 1017, 1058, 1173
<code>\Genplacrfullformat</code> 121, <u>123</u> , 261, 263–265, 271, 367, <u>368</u> , 378, 384, 456, 527–530, 535, 568, 575, 576, 580, 706, 707, 779, 781, 782, 786, 821, 828, 829, 833, 961, 1034–1037, 1041, 1115, 1116, 1190–1192, 1197	<code>\glossaries</code> 286–288
<code>\genplacrfullformat</code> 121, <u>122</u> , 123, 261, 263–265, 271, 367, <u>368</u> , 378, 384, 456, 527, 529, 530, 535, 568, 575, 576, 580, 706, 707, 779–782, 786, 821, 828, 829, 833, 960, 961, 1034–1036, 1041, 1115, 1116, 1190–1192, 1197	<code>\GlossariesWarning</code> 7, 9, 23, <u>23</u> , 24, 28, 29, 35, 37, 40, 49, 54, 58, 70, 74, 75, 81, 84, 88, 89, 115–117, 179, 194– 197, 200, 202–204, 207, 208, 214, 217, 218, 277, 350, 390– 392, 399, 400, 402, 403, 406– 409, 414, 417, 424, 426, 430, 431, 450, 457, 458, 485, 496, 497, 499–501, 503, 506, 508, 521, 523, 553, 589, 640, 642, 648, 649, 651–653, 656, 658, 659, 664, 667, 674, 676, 680, 681, 684, 700, 701, 707–709, 736, 747–749, 751–753, 755, 758–760, 773, 774, 802, 806, 842, 894, 896, 903, 906, 907, 910–912, 914, 917, 920, 922, 929, 932, 936, 937, 940, 957, 958, 991, 1001–1004, 1006– 1008, 1010, 1011, 1013, 1015, 1016, 1044, 1049, 1051, 1057, 1058, 1060–1062, 1065–1069, 1072, 1075, 1077, 1084, 1087, 1090, 1092, 1094, 1095, 1112, 1113, 1147, 1157–1159, 1161, 1163–1165, 1167, 1169, 1171, 1199
<code>\GetTitleStringDisableCommands</code> 292, 865	<code>\GlossariesWarningNoLine</code> . 7, 23, <u>23</u> , 44, 200, 201, 205, 221, 287, 390, 391, 399, 498, 499, 502,
<code>\GetTitleStringSetup</code> 292, 865	
<code>\glo@desc</code> 594, 847, 848	
<code>\glo@do@compare</code> . . 229, 230, 511, 512, 763, 764, 1019, 1174, 1175	
<code>\glo@grabfirst</code> 227, <u>237</u> , 514, 515, 766, 767, 1022, 1023, 1177, 1178	
<code>\glo@label</code> 71, 105, 424, 444, 674, 695, 930, 951, 1085, 1106	
<code>\glo@list</code> . . . 105, 444, 695, 951, 1106	
<code>\glo@name</code> . . 246, 520, 772, 1027, 1183	
<code>\glo@parent</code> . . . 71, 424, 674, 930, 1085	
<code>\glo@type</code> . . . 105, 444, 695, 951, 1106	

510, 610, 640, 641, 649, 750, 751, 754, 762, 863, 894, 895, 903, 1006, 1009, 1017, 1047, 1049, 1058, 1161, 1162, 1165, 1173	<code>\glossary</code> 39, 40, <u>40</u> , 408, 409, 590, 658, 659, 843, 913, 914, 1068, 1069	<code>\glossarysubentryfield</code> ... <u>248</u> , 377, 521, 574, 595–600, 602– 604, 773, 827, 848–853, 855–857
<code>\glossaryentry</code>	213, 214, 506, 507, 590, 757, 758, 843, 1013, 1014, 1168, 1169	<code>\glossarytitle</code> ... 186, 218, 219, 233, 236, 237, 241, 489, 508, 509, 514, 517, 592, 741, 760, 766, 769, 845, 995, 1016, 1021, 1024, 1151, 1171, 1172, 1177, 1180
<code>glossaryentry</code> (counter)	<u>14</u>	<code>\glossarytoctitle</code>
<code>\glossaryentryfield</code> <u>247</u> , 377, 520, 521, 574, 594–600, 602–604, 772, 773, 827, 847–853, 855–857		11, 19, 20, 41, 48, 50, 55, 186, 218, 233, 236, 237, 241, 242, 393, 397, 398, 409, 411, 413, 415, 489, 508, 514, 517, 592, 642, 647, 659, 661, 663, 665, 741, 760, 766, 769, 845, 897, 901, 902, 914, 915, 917, 919, 920, 995, 1016, 1021, 1024, 1025, 1051, 1056, 1069, 1072, 1073, 1075, 1151, 1171, 1177, 1180
<code>\glossaryentrynumbers</code>	11, 12, 187, 219, 220, 234, 236, 238, 239, 243, 393, 490, 509, 515, 516, 518, 592, 643, 741, 760, 761, 767, 768, 770, 845, 897, 996, 1016, 1017, 1022, 1023, 1025, 1026, 1052, 1151, 1172, 1178, 1179, 1181	<code>\glossentry</code>
<code>\glossaryheader</code>	187, 233, 236, <u>246</u> , 289, 292– 294, 296–300, 302, 303, 307– 313, 315–317, 319, 320, 322, 326, 328, 330, 333–338, 341, 489, 514, 520, 592, 611–618, 620–629, 631–639, 741, 766, 771, 845, 864, 866–871, 873– 882, 884–886, 888–892, 996, 1021, 1027, 1151, 1177, 1183	105, 219, 235, 238, <u>246</u> , 289, 293– 296, 298, 299, 307, 309, 310, 319, 321, 322, 326, 328, 330, 333, 335, 336, 338, 445, 509, 516, 521, 611–616, 620–622, 627–629, 631–636, 638, 695, 761, 767, 768, 773, 864, 866– 869, 873–875, 880–882, 884– 886, 888–891, 951, 1016, 1023, 1027, 1106, 1172, 1179, 1183
<code>\glossarymark</code>	55, <u>56</u> , 415, 665, 920, 921, 1075, 1076	<code>\Glossentrydesc</code> 139, <u>247</u> , 376, 520, 574, 772, 827, 1027, 1128, 1183
<code>\glossaryname</code> ...	19, <u>47</u> , 50, 397, 411–413, 646, 661–663, 901, 917–919, 1055, 1056, 1072, 1073	<code>\glossentrydesc</code>
<code>\glossarypostamble</code> <u>55</u> , 187, 233, 236, 414, 489, 514, 592, 664, 741, 766, 845, 920, 996, 1021, 1075, 1151, 1177	... 139, 247, <u>247</u> , 290, 293– 296, 298, 299, 307, 309, 311, 319, 321–323, 326, 328, 330, 333–336, 339, 340, 376, 520, 574, 611–616, 620–622, 627– 629, 631–633, 635–638, 772, 827, 864–869, 873–875, 880– 882, 884–892, 1027, 1128, 1183
<code>\glossarypreamble</code> <u>54</u> , 186, 233, 236, 414, 489, 514, 592, 664, 741, 766, 845, 920, 995, 1021, 1075, 1151, 1177	<code>\Glossentryname</code>
<code>\glossarysection</code>	55, 186, 233, 236, 237, 414, 489, 514, 592, 664, 741, 766, 845, 920, 995, 1021, 1075, 1151, 1177	139, <u>246</u> , 520, 772, 1027, 1128, 1183
<code>\glossarystyle</code> ...	<u>251</u> , 522, 523, 774	<code>\glossentryname</code>
<code>glossarysubentry</code> (counter)	<u>15</u>	... 139, 246, <u>246</u> , 248, 289, 290, 292–296, 298, 299, 307, 309, 310, 319, 321, 322, 326, 328, 330, 333–336, 339, 340, 376, 520, 574, 611–616, 620– 622, 627–629, 631–633, 635,

636, 638, 772, 827, 864–869,
 873–875, 880–882, 884–886,
 888–892, 1027, 1028, 1128, 1183
 \glossentrynameother 139, 1128
 \Glossentrysymbol
 139, 247, 377, 521,
 574, 772, 827, 1028, 1128, 1183
 \glossentrysymbol 139, 247, 247,
 289, 290, 299, 311, 322, 323,
 330, 333–336, 339, 340, 377,
 520, 574, 611, 612, 616, 622,
 629, 633, 635, 636, 638, 772,
 827, 864, 865, 869, 875, 882,
 886–892, 1027, 1028, 1128, 1183
 \GLS 142, 469, 720, 974, 1129
 \Gls 115, 141, 142,
 255, 274, 450, 468, 525, 537,
 701, 720, 776, 788, 958, 973,
 1031, 1043, 1112, 1129, 1187, 1199
 \gls 40, 115, 141,
 142, 200, 245, 255, 274, 408,
 450, 468, 499, 519, 525, 537,
 658, 700, 719, 751, 771, 776,
 777, 788, 914, 957, 973, 1006,
 1026, 1031, 1043, 1068, 1112,
 1128, 1129, 1162, 1182, 1187, 1199
 \gls@delimN . . . 253, 1030, 1185, 1186
 \gls@accessibility 351, 351, 554, 807
 \gls@accsupp@engine 351, 351, 554, 807
 \gls@Alphpage
 207, 208, 212, 503, 505,
 755, 757, 1010, 1012, 1166, 1168
 \gls@alphpage
 207, 207, 212, 503, 505,
 755, 757, 1010, 1012, 1166, 1168
 \gls@arabicpage
 207, 208, 212, 503, 505,
 755, 756, 1010–1012, 1166, 1168
 \gls@assign@desc
 98, 98, 103, 440, 443, 690,
 691, 694, 946, 947, 950, 1101, 1105
 \gls@assign@desc@field 24
 \gls@assign@descplural 537, 544,
 545, 547–549, 583, 585–587,
 789, 797, 799–801, 836, 839, 840
 \gls@assign@descplural@field . . 25
 \gls@assign@field
 86, 86, 87, 92, 93, 98,
 100, 102, 103, 343, 344, 433,
 436, 440–444, 605, 683, 686,
 687, 690, 692–694, 858, 939,
 942, 943, 946, 948–950, 1094,
 1097, 1098, 1101, 1102, 1104, 1105
 \gls@assign@firstpl . . . 537–545,
 547–549, 583–587, 789, 791,
 793, 795, 797, 799–801, 836–840
 \gls@assign@name@field 25
 \gls@assign@plural
 537–539, 541–545,
 547–549, 583–587, 789, 791,
 793, 795, 797, 799–801, 836–840
 \gls@assign@symbolplural
 538, 539, 541–
 543, 547–549, 583–587, 789,
 791–793, 795, 799–801, 836–840
 \gls@assign@type@field 24
 \gls@automake@makegloss
 37, 37, 38, 197,
 912, 913, 1004, 1066–1068, 1160
 \gls@automake@nr 37, 37,
 38, 45, 197, 407, 497, 657, 749,
 912, 913, 1004, 1066–1068, 1160
 \gls@automake@option
 37, 37, 38, 912, 913, 1066–1068
 \gls@automake@val 37,
 38, 407, 657, 912, 913, 1066–1068
 \gls@begindocdefs
 89, 89, 434, 684, 940, 1095
 \gls@checkisacronymlist
 22, 126, 398,
 459, 648, 709, 902, 963, 1057, 1118
 \gls@checkseeallowed 81,
81, 199, 201, 429, 430, 433,
 498, 499, 679, 680, 750, 751,
 935, 1005, 1006, 1090, 1161, 1162
 \gls@checkseeallowed@preambleonly
 81, 430, 433, 680, 936, 1090
 \gls@codepage 34, 66, 195,
 196, 221, 406, 422, 495, 496,
 510, 655, 672, 747, 748, 762,
 910, 928, 1002, 1003, 1017,
 1065, 1082, 1083, 1157–1159, 1173
 \gls@debug@nr 7,
 390, 408, 409, 640, 659, 894, 1049
 \gls@debug@val
 7, 390, 408, 640, 659, 894, 1049
 \gls@defdocnewglossaryentry
 87, 89, 112,
 433, 434, 448, 683, 684, 698,
 939, 940, 955, 1094, 1095, 1110

<code>\gls@defglossaryentry</code>	<code>\gls@inlinesubsep</code>
..... 87–89, 98, <u>98</u> , 433,	289,
434, 440, 683–685, 691, 939,	290, 595, 611, 612, 848, 864, 865
940, 947, 1094, 1095, 1101, 1102	<code>\gls@islistofacronyms</code>
<code>\gls@delimN</code> 252, <u>253</u> , 1030, 1185 21, 398, 648, 902, 1057
<code>\gls@delimR</code> 252, <u>252</u> , 1029, 1030, 1185	<code>\gls@istfilebase</code>
<code>\gls@disablepagerefexpansion</code> ..	51, 52, <u>52</u> , 195, 196, 413, 414,
206, <u>207</u> , 212, 503, 505, 755,	495, 496, 663, 664, 747, 748,
757, 1010, 1012, 1165, 1166, 1168	919, 1002, 1003, 1074, 1158, 1159
<code>\gls@do@addxdyattribute</code>	<code>\gls@kernelglossredefs@nr</code>
..... 60, 418, 668, 924, 1078 40, 914, 1068
<code>\gls@docdef@addlabel</code>	<code>\gls@kernelglossredefs@val</code>
87, 40, 914, 1068
88, 683, 684, 939, 940, 1094, 1095	<code>\gls@label</code>
<code>\gls@docdef@labellist</code> 255, 524, 525, 776, 1031, 1186
87,	<code>\gls@level</code>
<u>87</u> , 88, 683, 684, 939, 940, 1094	85, 101, 102,
<code>\gls@doclearpage</code>	234, 238, 432, 441, 442, 515,
57, <u>57</u> , 416, 666, 922, 1076, 1077	682, 692, 693, 767, 938, 948,
<code>\gls@doentryfmt</code> <u>75</u> , 426, 676, 932, 1087	949, 1022, 1092, 1103, 1104, 1178
<code>\gls@dostubst</code>	<code>\gls@local@stepcounter</code>
.. 131, 461, 712, 966, 1120, 1121 <u>107</u> , 108, 953, 1107, 1108
<code>\gls@dotocitle</code>	<code>\gls@measuring@reset</code>
218, 219, 241, 509, 517, 760, 107, 108, 953, 1107, 1108
761, 769, 1016, 1024, 1172, 1180	<code>\gls@mfirstuc@nr</code>
<code>\gls@end@sanitizesort</code>	42, 915, 1070
25,	<code>\gls@mfirstuc@val</code> 42, 915, 1070
26, 400, 401, 650, 904, 905, 1059	<code>\gls@newacronym</code> 255, <u>256</u> , 1031, 1187
<code>\gls@endcheck</code>	<code>\gls@noidx@nonumberlist</code>
86,	.. <u>82</u> , 203, 936, 1008, 1091, 1164
432, 433, 682, 683, 938, 939, 1093	<code>\gls@noidxglossary</code>
<code>\gls@glossary</code> 201, <u>204</u> , 499, 502,
.. 40, 41, <u>205</u> , 213, 214, 408,	751, 753, 1006, 1008, 1162, 1164
409, 502, 505, 506, 658, 659,	<code>\gls@nonumberlist@nr</code>
754, 757, 758, 914, 1009, 1013, 82, 430, 680, 936, 1090
1014, 1068, 1069, 1165, 1168, 1169	<code>\gls@nonumberlist@val</code>
<code>\gls@glsdefs@addlabel</code> 82, 430, 680, 936, 1090
87,	<code>\gls@nosetquote</code>
89, 683, 685, 939, 940, 1094, 1095	99, 189,
<code>\gls@glsdefs@labellist</code> 87,	190, 193, 440, 490, 492, 494,
<u>87</u> , 88, 683, 684, 939, 940, 1094	691, 742, 743, 746, 947, 997,
<code>\gls@gobbleopt</code>	998, 1001, 1102, 1152, 1154, 1157
75, 77, 427, 677, 933, 1087, 1088	<code>\gls@number</code>
<code>\gls@grplabel</code> ... 285, 609, 863, 1046	.. 211, 505, 756, 1012, 1167, 1168
<code>\gls@hypergrouprerun</code>	<code>\gls@numberedsection@nr</code>
..... 287, <u>287</u> , 610, 863, 1047 10, 241, 393, 517,
<code>\gls@if@measuring</code>	642, 769, 896, 1024, 1051, 1180
107, <u>107</u> , 108, 952, 953, 1107, 1108	<code>\gls@numberedsection@val</code>
<code>\gls@ifnotmeasuring</code> ... <u>107</u> , 109, 10, 241, 393, 517,
110, 207, 446, 447, 696, 697,	642, 769, 896, 1024, 1051, 1180
953, 954, 1010, 1107, 1109, 1166	<code>\gls@numberpage</code>
<code>\gls@inlinepostchild</code> 207, <u>208</u> , 211, 503, 505,
289,	755, 756, 1010, 1012, 1166, 1167
290, 595, 611, 612, 848, 864, 865	
<code>\gls@inlinesep</code>	
289, 290,	
594, 595, 611, 847, 848, 864, 865	

<code>\gls@org@glossaryentryfield</code> . . .	<code>\gls@seenoindex@val</code>
. 219, 509, 761, 1016, 1172 9, 392, 642, 896, 1050
<code>\gls@org@glossarysubentryfield</code>	<code>\gls@set@xr@key</code>
. . 219, 220, 509, 761, 1016, 1172 81, 81, 429, 679, 935, 1090
<code>\gls@org@insert</code>	<code>\gls@start@measuring</code>
. 542–544, 546, 794, 796, 798 108, 108, 109, 953, 1108
<code>\gls@org@label</code>	<code>\gls@stop@measuring</code>
. . 107, 108, 952, 953, 1107, 1108 108, 108, 109, 953, 1108
<code>\gls@org@link</code> 108, 953, 1108	<code>\gls@suffixF</code> 53, 53,
<code>\gls@org@refstepcounter</code>	188, 190, 414, 490, 492, 592,
107, 107, 108, 952, 953, 1107, 1108	594, 664, 741, 743, 845, 847,
<code>\gls@org@target</code> 108, 953, 1108	920, 996, 998, 1074, 1152, 1153
<code>\gls@orgAlph</code> 211, 212, 504,	<code>\gls@suffixFF</code> 53, 53,
505, 756, 757, 1012, 1167, 1168	188, 190, 414, 490, 492, 592,
<code>\gls@orgalph</code> 211, 212, 504,	594, 664, 741–743, 845, 847,
505, 756, 757, 1012, 1167, 1168	920, 996, 998, 1074, 1152–1154
<code>\gls@orgarabic</code> 211, 212,	<code>\gls@text</code>
504, 505, 756, 1012, 1167, 1168	122, 123, 456, 457, 707, 961, 1116
<code>\gls@orgnumber</code>	<code>\gls@the</code> . . . 211, 505, 756, 1012, 1167
. . 211, 504, 505, 756, 1012, 1167	<code>\gls@thissty</code> . . 31, 405, 654, 909, 1063
<code>\gls@orgRoman</code> 211, 212,	<code>\gls@tmp</code> 197, 199, 205, 277, 502, 545,
505, 756, 757, 1012, 1167, 1168	754, 798, 1004, 1005, 1009,
<code>\gls@orgromannumeral</code> 211, 212, 504,	1044, 1160, 1161, 1164, 1165, 1199
505, 756, 757, 1012, 1167, 1168	<code>\gls@tmplen</code> . . . 138, 337, 339, 340,
<code>\gls@orgthe</code>	467, 600, 601, 637, 638, 718,
. . 211, 504, 505, 756, 1012, 1167	719, 853, 854, 891, 892, 972, 1127
<code>\gls@original@glossary</code>	<code>\gls@tr@set@acronym@toctitle</code> . .
. 39, 40, 408,	20, 397, 398, 647, 901, 902, 1056
409, 658, 659, 913, 914, 1068, 1069	<code>\gls@tr@set@main@toctitle</code>
<code>\gls@original@makeglossary</code> 19, 397, 646, 901, 1056
. 39, 41, 408,	<code>\gls@tr@set@numbers@toctitle</code> . .
409, 658, 659, 913, 914, 1068, 1069 41, 409, 659, 915, 1069
<code>\gls@printglossary@begin</code>	<code>\gls@tr@set@symbols@toctitle</code> . .
. 219, 220, 1172 41, 409, 659, 914, 1069
<code>\gls@printglossary@end</code> 220, 220, 1172	<code>\gls@translate@nr</code>
<code>\gls@protected@pagefmts</code> 131, 207,	30, 31, 404, 653, 908, 1062, 1063
207, 208, 461, 503, 504, 712,	<code>\gls@translate@val</code>
755, 966, 1010, 1011, 1120, 1166 30, 404, 653, 908, 1062
<code>\gls@Romanpage</code>	<code>\gls@wrglossary</code>
. 207, 208, 212, 503–505,	. . 206, 206, 503, 754, 1010, 1165
755, 757, 1010–1012, 1166, 1168	<code>\gls@xdy@set@lang</code>
<code>\gls@romanpage</code>	34, 195, 196, 220, 910, 1002,
. 207, 208, 212, 503–505,	1003, 1017, 1064, 1157, 1158, 1172
755, 757, 1010–1012, 1166, 1168	<code>\gls@xdystring</code> 130,
<code>\gls@save@numberlist</code>	131, 461, 712, 965, 966, 1120, 1121
. 11, 12, 216, 393, 507,	<code>\gls@xindy@glsnumbersfalse</code>
643, 759, 897, 1015, 1052, 1170 35, 406, 656, 910, 1065
<code>\gls@seenoindex@nr</code>	<code>\gls@xindy@glsnumberstrue</code>
. 9, 392, 642, 896, 1050 34, 406, 655, 910, 1064

<code>\gls@xr@key</code> . . .	9, 81, <u>81</u> , 392, 429, 430, 641, 642, 679, 680, 895, 896, 935, 936, 1050, 1051, 1090	<code>\glsautomaketrue</code>	37, 38, 407, 657, 912, 913, 1067, 1068
<code>\glsaccessdisplay</code>	<u>363</u> , 561, 814	<code>\glsautoprefix</code>	<u>10</u> , 11, 242, 393, 517, 642, 643, 769, 896, 897, 1025, 1051, 1180
<code>\glsaccessibility</code> <u>351</u> , 359, 554, 559, 807, 812	<code>\glsbackslash</code> 45, <u>183</u> , 487, 738, 993, 1148
<code>\glsaccsupp</code> . . .	359, <u>359</u> , 360, 559, 812	<code>\glscapitalisewords</code> 175, <u>175</u> , 734, 735, 989, 1145
<code>\glsacronymtrue</code>	20, 398, 647, 901, 1056	<code>\glscapspace</code>	118–121, 141–146, 162–169, 268, 269, 363, 364, 366, 367, 369–371, 381, 451, 452, 454–456, 468– 470, 476–481, 533, 542, 546, 562, 563, 565–570, 577, 578, 701, 703–707, 719–722, 728– 732, 784, 785, 794, 799, 815, 816, 818–823, 831, 958–961, 973–975, 982–986, 1039, 1113– 1116, 1129–1131, 1137–1142, 1195
<code>\glsacrpluralsuffix</code>	48, <u>48</u> , 256, 266, 270–272, 412, 525, 530, 531, 534–536, 538, 662, 777, 782, 783, 786, 787, 790, 917, 1031, 1037, 1041, 1042, 1072, 1187, 1193, 1196–1198	<code>\glscategory</code>	358, 359, 559, 812
<code>\glsacrshortcutsfalse</code> 46, 410, 660, 916, 1071	<code>\glsclearpage</code>	56, 57, 416, 666, 921, 922, 1076, 1077
<code>\glsacrshortcutstrue</code> 46, 410, 660, 916, 1070	<code>\glsclosebrace</code>	63, 64, <u>183</u> , 187, 188, 420, 487, 490, 592, 670, 738, 741, 742, 845, 846, 926, 993, 996, 1081, 1148, 1152
<code>\glsacspace</code> 264, <u>264</u> , 267, 529–531, 781, 783, 1036, 1038, 1191, 1192, 1194	<code>\glscompositor</code>	52, 52, 63, 190, 414, 420, 491, 591, 594, 664, 670, 743, 844, 847, 919, 926, 998, 1074, 1080, 1153
<code>\glsadd</code> 40, 181, <u>181</u> , 182, 408, 486, 658, 738, 914, 992, 993, 1068, 1148		<code>\glscounter</code>	22, <u>22</u> , 46, 59, 77, 102, 128, 399, 411, 417, 427, 443, 459, 588, 648, 661, 667, 677, 693, 710, 842, 902, 916, 923, 933, 949, 964, 1057, 1071, 1077, 1088, 1104, 1119
<code>\glsaddall</code> . .	<u>182</u> , 486, 738, 992, 1148	<code>\glscurrententrylabel</code>	139, 216, 219, 220, 507, 509, 759, 761, 1015, 1016, 1127, 1170, 1172
<code>\glsaddallunused</code> <u>182</u> , 486, 738, 993, 1148	<code>\glscurrentfieldvalue</code> . . .	73, 74, <u>74</u> , 426, 676, 931, 932, 1086, 1087
<code>\glsaddkey</code> . . .	<u>92</u> , 436, 686, 942, 1097	<code>\glscustomtext</code> . . .	118, 120, 122, 141–146, 162–169, 268, 269, 363, 366, 368–371, 381, 382, 451, 453, 455, 456, 468–470, 476–481, 532, 533, 538, 539, 541–544, 546, 562, 564–570, 577, 578, 701, 704, 706, 707, 719–722, 728–732, 784, 785,
<code>\GlsAddLetterGroup</code> <u>67</u> , 423, 673, 928, 1083		
<code>\glsaddprotectedpagefmt</code> . .	208, <u>208</u> , 504, 755, 1011, 1166, 1167		
<code>\GlsAddSortRule</code>	<u>65</u> , 421, 671, 927, 1082		
<code>\glsaddstoragekey</code> <u>91</u> , 435, 686, 942, 1097		
<code>\GlsAddXdyAlphabet</code> <u>61</u> , 419, 669, 925, 1079		
<code>\GlsAddXdyAttribute</code>	59, <u>59</u> , 61, 417–419, 588, 589, 667– 669, 841, 842, 923, 924, 1078, 1079		
<code>\GlsAddXdyCounters</code> 59, <u>59</u> , 60, 77, 417, 418, 428, 589, 667, 668, 678, 842, 923, 924, 934, 1077–1079, 1088		
<code>\GlsAddXdyLocation</code>	<u>63</u> , 420, 421, 589, 670, 671, 842, 926, 1081		
<code>\GlsAddXdyStyle</code> <u>65</u> , 421, 422, 671, 672, 927, 1082		
<code>\glsautomakefalse</code>	37, 38, 197, 407, 497, 657, 658, 749, 912, 913, 1004, 1067, 1160		

790, 791, 794, 796, 798, 799,
815, 817, 818, 820–823, 830,
831, 958, 960, 961, 973–975,
982–986, 1039, 1040, 1113–
1116, 1129–1131, 1137–1142, 1195
\GlsDeclareNoHyperList
. [22](#), [23](#), [399](#), [648](#), [903](#), [1057](#)
\glsdefaultshortaccess [386](#),
[387](#), [582–584](#), [586](#), [835–837](#), [839](#)
\glsdefaultttype [19](#),
[19](#), [20](#), [34](#), [54](#), [66](#), [67](#), [74–76](#),
[100](#), [117](#), [205](#), [217](#), [218](#), [397](#),
[414](#), [422](#), [423](#), [426](#), [427](#), [441](#),
[450](#), [457](#), [502](#), [508](#), [647](#), [664](#),
[672](#), [673](#), [676](#), [677](#), [692](#), [701](#),
[707](#), [708](#), [754](#), [760](#), [901](#), [910](#),
[920](#), [927](#), [928](#), [932](#), [933](#), [948](#),
[958](#), [1009](#), [1015](#), [1016](#), [1056](#),
[1064](#), [1065](#), [1075](#), [1082](#), [1083](#),
[1087](#), [1088](#), [1102](#), [1113](#), [1165](#), [1171](#)
\glsdefmain
[19](#), [20](#), [78](#), [397](#), [428](#), [646](#), [647](#),
[678](#), [901](#), [934](#), [1055](#), [1056](#), [1088](#)
\glsdefs@newdocentry
. [89](#), [89](#), [684](#), [685](#), [940](#), [941](#), [1095](#)
\GLSdesc [152](#), [473](#), [724](#), [978](#), [1133](#), [1134](#)
\Glsdesc [152](#), [152](#), [473](#), [724](#), [978](#), [1133](#)
\glsdesc [151](#), [152](#), [473](#), [724](#), [978](#), [1133](#)
\GLSdescplural [153](#), [473](#), [725](#), [978](#), [1134](#)
\Glsdescplural
. [153](#), [153](#), [473](#), [725](#), [978](#), [1134](#)
\glsdescplural
. [152](#), [153](#), [473](#), [725](#), [978](#), [1134](#)
\glsdescriptionaccessdisplay
. [361](#), [373](#), [376](#), [377](#),
[560](#), [563–565](#), [571](#), [572](#), [574](#),
[813](#), [816–818](#), [824](#), [825](#), [827](#), [828](#)
\glsdescriptionpluralaccessdisplay
. [361](#), [373](#), [560](#),
[562](#), [563](#), [572](#), [813](#), [815](#), [816](#), [825](#)
\glsdescwidth
[296](#), [296](#), [297](#), [298](#), [301](#), [304](#),
[305](#), [307](#), [307](#), [308–312](#), [318](#),
[319–322](#), [324–326](#), [326](#), [327–](#)
[331](#), [614](#), [615](#), [617–623](#), [626–](#)
[628](#), [630–634](#), [867–876](#), [880–887](#)
\glsdetoklabel [36](#),
[69](#), [69](#), [70–73](#), [82](#), [83](#), [89](#), [90](#),
[94–99](#), [105](#), [110–114](#), [127](#), [170](#),
[172](#), [180](#), [181](#), [201–203](#), [210](#),
[212](#), [219](#), [220](#), [228–230](#), [233–](#)
[235](#), [238](#), [240](#), [244](#), [246](#), [279–](#)
[283](#), [338](#), [344](#), [345](#), [360–363](#),
[387–389](#), [423–426](#), [430](#), [434](#),
[438–440](#), [444](#), [447–449](#), [459](#),
[481](#), [485](#), [486](#), [499–501](#), [504](#),
[505](#), [509](#), [511](#), [512](#), [515](#), [517](#),
[519](#), [520](#), [551–554](#), [559–561](#),
[587](#), [588](#), [605](#), [606](#), [637](#), [657](#),
[673–676](#), [680](#), [685](#), [688–691](#),
[695](#), [698–700](#), [710](#), [732](#), [733](#),
[737](#), [738](#), [751](#), [752](#), [756](#), [757](#),
[761](#), [763](#), [764](#), [767](#), [768](#), [770–](#)
[772](#), [804–807](#), [812–814](#), [840](#),
[841](#), [859](#), [891](#), [911](#), [929–932](#),
[936](#), [941](#), [944–947](#), [951](#), [954–](#)
[957](#), [963](#), [986](#), [987](#), [991](#), [992](#),
[1006–1008](#), [1012](#), [1016](#), [1018](#),
[1019](#), [1022–1024](#), [1026](#), [1027](#),
[1044–1046](#), [1066](#), [1084–1086](#),
[1091](#), [1095](#), [1099–1102](#), [1106](#),
[1109–1112](#), [1118](#), [1142](#), [1143](#),
[1147](#), [1148](#), [1162](#), [1163](#), [1167](#),
[1168](#), [1172](#), [1174](#), [1175](#), [1178](#),
[1180](#), [1182](#), [1183](#), [1200](#), [1201](#)
\glsdisablehyper
. [140](#), [467](#), [719](#), [973](#), [1128](#)
\Glsdisp [146](#), [146](#), [975](#), [1131](#)
\glsdisp [145](#), [146](#), [470](#), [722](#), [975](#), [1131](#)
\glsdisplay [123](#), [451](#), [457](#), [701](#), [707](#), [708](#)
\glsdisplayfirst
. [123](#), [451](#), [457](#), [701](#), [707](#), [708](#)
\glsdisplaynumberlist
. [179](#), [202](#), [485](#), [500](#), [736](#),
[737](#), [752](#), [991](#), [1007](#), [1147](#), [1163](#)
\glsdohyperlink [139](#),
[140](#), [467](#), [468](#), [719](#), [972](#), [973](#), [1128](#)
\glsdohyperlinkhook [139](#), [139](#), [1128](#)
\glsdohypertarget [138](#), [140](#), [467](#),
[468](#), [718](#), [719](#), [972](#), [973](#), [1127](#), [1128](#)
\glsdohypertargethook [138](#), [138](#), [1127](#)
\glsdoifexists
. [69](#), [71](#), [73](#), [94–97](#), [109](#), [110](#),
[141–146](#), [162–169](#), [175](#), [179](#),
[181](#), [202](#), [203](#), [223](#), [346–349](#),
[369](#), [370](#), [376](#), [377](#), [423–425](#),
[438](#), [439](#), [446](#), [447](#), [468–471](#),
[476–481](#), [483](#), [485](#), [486](#), [500](#),
[501](#), [569](#), [570](#), [574](#), [606–609](#),
[673–675](#), [688–690](#), [697](#), [719–](#)

722, 728–732, 735–738, 752,
753, 822, 823, 827, 860–862,
929–931, 944–946, 954, 973–
975, 982–986, 989, 991, 992,
1007, 1008, 1084–1086, 1099–
1101, 1109, 1129–1131, 1137–
1142, 1145, 1147, 1148, 1163
\glsdoifexistsordo . 70, 126, 170,
424, 458, 481, 674, 709, 732,
929, 962, 986, 1084, 1117, 1142
\glsdoifexistsorwarn
. 70, 240, 246, 247, 424, 517,
520, 521, 674, 768, 772, 929,
1024, 1027, 1028, 1084, 1179, 1183
\glsdoifnoexists 69, 87, 98,
423, 433, 440, 673, 683, 690,
929, 939, 946, 1084, 1094, 1101
\glsdonohyperlink
. 128, 139, 140, 459, 467,
710, 719, 964, 972, 973, 1118, 1128
\glsdoparenifnotempty
. 277, 545, 798, 1044, 1199
\glsdosanitizesort
. 16, 395, 645, 899, 1054
\glsdoshowtarget 8, 138,
139, 895, 972, 1050, 1127, 1128
\glsenableentrycount . . 112, 448,
449, 698, 700, 955, 957, 1110, 1112
\glsenablehyper
. 140, 468, 719, 973, 1128
\glsencapwrcontent
. 128, 128, 964, 1118, 1119
\glsentryaccess
. 356, 360, 557, 559, 810, 813
\glsentrycounter . 128, 250, 253,
459, 522, 524, 710, 774, 776,
964, 1029, 1030, 1119, 1184, 1186
\glsentrycounterfalse
. 14, 395, 644, 898, 1053
\glsentrycounterlabel 245,
245, 519, 771, 1026, 1027, 1182
\GlsEntryCounterLabelPrefix
. 242, 244, 518, 519,
769–771, 1025, 1026, 1181, 1182
\glsentrycountertrue
. 14, 395, 644, 898, 1053
\glsentrycurrcount 112, 114, 448,
449, 699, 700, 956, 957, 1110, 1112
\Glsentrydesc . 152, 173, 173, 247,
373, 377, 473, 482, 520, 571,
574, 724, 734, 772, 824, 827,
978, 988, 1027, 1133, 1144, 1183
\glsentrydesc . 139, 152, 173, 247,
373, 376, 452, 453, 473, 482,
520, 563–565, 571, 572, 574,
703, 704, 724, 734, 772, 816–
818, 824, 825, 827, 978, 988,
1027, 1128, 1133, 1134, 1144, 1183
\glsentrydescaccess
. 357, 361, 558, 560, 811, 813
\Glsentrydescplural 153,
173, 173, 373, 473, 482, 572,
725, 734, 825, 978, 988, 1134, 1144
\glsentrydescplural . . . 153, 173,
373, 451, 452, 473, 482, 562,
563, 572, 702, 703, 725, 734,
815, 816, 825, 978, 988, 1134, 1144
\glsentrydescpluralaccess
. 357, 361, 558, 560, 811, 813
\Glsentryfirst 116,
119, 148, 174, 174, 365, 371,
450, 453, 455, 471, 483, 564,
566, 571, 701, 704, 705, 723,
734, 817, 819, 824, 958, 959,
976, 989, 1112, 1114, 1132, 1144
\glsentryfirst
. 115, 119, 120, 148, 174,
365, 371, 450, 452–455, 471,
472, 483, 563, 564, 566, 571,
700, 703–705, 723, 734, 816,
817, 819, 824, 957, 959, 960,
976, 989, 1112, 1114, 1132, 1144
\glsentryfirstaccess
. 356, 361, 557, 560, 811, 813
\Glsentryfirstplural 117,
118, 150, 175, 175, 364, 372,
450, 452, 454, 472, 483, 562,
565, 571, 701, 702, 705, 724,
734, 815, 818, 824, 958, 959,
977, 989, 1113, 1114, 1133, 1145
\glsentryfirstplural
. 116, 118, 119, 150,
175, 364, 372, 450–452, 454,
472, 483, 562, 563, 565, 566,
571, 701–705, 724, 734, 815,
816, 818, 819, 824, 958, 959,
977, 989, 1113, 1114, 1133, 1144
\glsentryfirstpluralaccess
. 356, 361, 558, 560, 811, 813

<code>\glsentryfmt</code>	77, 78, 118 , 427, 428, 451, 677, 678, 701, 933, 934, 958, 1088, 1089, 1113	984, 985, 987, 990, 991, 1035, 1036, 1038–1043, 1112, 1116, 1140, 1141, 1143, 1146, 1191–1199
<code>\Glsentryfull</code>	179, 179 , 261, 270, 271, 383, 385, 485, 527, 534, 535, 579, 581, 736, 779, 786, 787, 832, 834, 991, 1034, 1040, 1042, 1146, 1190, 1196, 1197	<code>\glsentrylongaccess</code> 357 , 362, 558, 561, 811, 814
<code>\glsentryfull</code>	178 , 261, 270, 271, 382, 385, 485, 527, 534, 535, 579, 581, 736, 779, 785, 787, 832, 834, 991, 1034, 1040, 1042, 1146, 1190, 1196, 1197	<code>\Glsentrylongpl</code> 117, 168, 178, 178 , 179, 263, 264, 268, 270, 368, 378, 381–383, 450, 480, 484, 485, 529, 533, 534, 568, 575, 578, 579, 701, 732, 736, 781, 784–786, 822, 828, 831, 832, 958, 986, 991, 1035, 1036, 1039, 1040, 1113, 1141, 1146, 1191, 1192, 1195, 1196
<code>\Glsentryfullpl</code> 179, 179 , 261, 270, 272, 383, 385, 485, 528, 534, 535, 579, 581, 736, 779, 786, 787, 832, 834, 991, 1034, 1040, 1042, 1146, 1147, 1190, 1196, 1198	<code>\glsentrylongpl</code> 116, 122, 168, 169, 178 , 179, 263– 265, 268–272, 368, 378, 379, 381–385, 450, 456, 480, 481, 484, 485, 529, 530, 533–535, 544, 550, 568, 575, 576, 578– 581, 701, 707, 731, 732, 736, 780–782, 784–787, 796, 803, 821, 828, 829, 831–834, 958, 961, 985, 986, 991, 1035–1037, 1039–1042, 1113, 1116, 1141, 1142, 1146, 1191, 1192, 1195–1198	<code>\glsentryfullpl</code> 179 , 261, 270, 272, 383, 385, 485, 527, 534, 535, 579, 581, 736, 779, 786, 787, 832, 834, 991, 1034, 1040, 1042, 1146, 1190, 1196, 1197
<code>\glsentryitem</code> 245 , 289, 292– 296, 298, 299, 307, 309, 310, 319, 321, 322, 326, 328, 330, 333, 335, 336, 339, 519, 594– 600, 602–604, 611–616, 620– 622, 627–629, 631–633, 635, 636, 638, 771, 848–853, 855– 857, 864–869, 873–875, 880– 882, 884–886, 888–891, 1026, 1182	<code>\Glsentrylong</code> 116, 166, 172, 178 , 178 , 179, 263, 264, 269, 270, 368, 370, 378, 381–383, 450, 479, 482, 484, 485, 529, 533, 534, 568, 570, 575, 578, 579, 701, 731, 733, 736, 780, 781, 785, 786, 821, 823, 828, 831, 832, 958, 985, 987, 990, 991, 1035, 1036, 1040, 1112, 1140, 1143, 1146, 1191, 1195, 1196	<code>\glsentrylongpluralaccess</code> 357 , 362, 558, 561, 811, 814
<code>\Glsentrylong</code> 115, 122, 166, 167, 172, 178, 178 , 263–273, 368, 370, 371, 377– 386, 450, 456, 479, 480, 482, 484, 485, 529–536, 544, 568, 570, 575–582, 700, 707, 730, 731, 733, 736, 780–788, 796, 821, 823, 828–835, 957, 961,	<code>\glsentryname</code> 139, 151, 171 , 216, 337, 372, 376, 472, 473, 481, 571, 574, 637, 724, 733, 824, 827, 891, 977, 978, 987, 1015, 1127, 1128, 1133, 1143, 1170	<code>\Glsentryname</code> . 151, 171, 171 , 246, 372, 376, 473, 481, 520, 571, 574, 724, 733, 772, 824, 827, 977, 987, 1027, 1133, 1143, 1183
	<code>\glsentrynumberlist</code> 179, 179 , 202, 485, 500, 736, 737, 752, 991, 1007, 1147, 1162	<code>\glsentryparent</code> 176 , 483, 735, 989, 1145
	<code>\Glsentryplural</code> 118, 149, 174, 174 , 364, 372, 451, 454, 472, 482, 562, 565, 571, 702, 705, 723, 734, 815, 818, 824, 959, 977, 988, 1114, 1132, 1144	<code>\glsentryplural</code> 118, 119, 149, 173 , 363, 364, 372, 451, 452, 454, 472, 482, 562, 563,

	565, 571, 702, 704, 705, 723, 734, 815, 816, 818, 824, 959, 977, 988, 1113, 1114, 1132, 1144		581, 706, 730, 736, 782, 786, 787, 820, 829, 833, 834, 960, 983, 990, 1037, 1041, 1042, 1115, 1139, 1146, 1192, 1197, 1198
<code>\glsentrypluralaccess</code>	356, 360, 558, 560, 811, 813	<code>\glsentryshortpl</code>	120, 122, 164, 165, 178, 179, 263–265, 270–272, 366, 368, 378, 382–385, 455, 456, 478, 484, 485, 529, 530, 534, 535, 550, 567–569, 575, 576, 579– 581, 706, 707, 729, 730, 736, 781, 782, 785–787, 803, 820– 822, 828, 829, 832–834, 960, 961, 983, 984, 990, 991, 1035, 1036, 1040–1042, 1115, 1116, 1139, 1146, 1191, 1192, 1196, 1197
<code>\Glsentryprefix</code>	344, 345, 347, 606, 607, 859, 861	<code>\glsentryshortpluralaccess</code>	357, 362, 558, 560, 811, 814
<code>\glsentryprefix</code>	344, 345, 346, 349, 605, 607, 609, 859, 860, 862	<code>\glsentrysort</code> 176, 483, 735, 989, 1145	
<code>\Glsentryprefixfirst</code>	344, 344, 347, 605, 608, 859, 861	<code>\Glsentrysymbol</code>	154, 174, 174, 247, 373, 377, 474, 483, 521, 572, 574, 725, 734, 772, 825, 827, 979, 988, 989, 1028, 1134, 1144, 1183
<code>\glsentryprefixfirst</code> 344, 344, 346, 349, 605, 607, 609, 859, 860, 862		<code>\glsentrysymbol</code>	139, 154, 174, 247, 373, 377, 452, 453, 473, 474, 482, 521, 539, 542, 546, 563–565, 572, 574, 703, 704, 725, 734, 772, 791, 794, 799, 816–818, 825, 827, 979, 988, 989, 1028, 1128, 1134, 1144, 1183
<code>\Glsentryprefixfirstplural</code>	344, 344, 348, 606, 608, 859, 861	<code>\glsentrysymbolaccess</code>	357, 361, 558, 560, 811, 813
<code>\glsentryprefixfirstplural</code>	344, 344, 347, 350, 605, 607, 609, 859, 860, 862	<code>\Glsentrysymbolplural</code>	155, 174, 174, 374, 474, 483, 572, 725, 734, 825, 979, 989, 1135, 1144
<code>\Glsentryprefixplural</code>	345, 345, 348, 606, 608, 859, 861	<code>\glsentrysymbolplural</code>	155, 174, 374, 451, 452, 474, 483, 538, 542, 546, 562, 563, 572, 702, 703, 725, 726, 734, 791, 794, 799, 815, 816, 825, 979, 989, 1135, 1144
<code>\glsentryprefixplural</code>	344, 345, 347, 350, 605, 607, 609, 859, 860, 862	<code>\glsentrysymbolpluralaccess</code>	357, 361, 558, 560, 811, 813
<code>\glsentryprevcount</code> . 112–114, 448, 449, 699, 700, 956, 957, 1110, 1111		<code>\Glsentrytext</code>	119, 147, 173, 173, 365, 371, 452, 454, 471, 482, 563, 566, 570, 703, 705, 723, 734, 817, 819, 823, 959, 976, 988, 1114, 1132, 1144
<code>\Glsentryshort</code>	121, 162, 172, 178, 178, 265, 271, 367– 369, 378, 384, 385, 455, 477, 482, 484, 530, 534, 535, 567– 569, 575, 580, 581, 706, 728, 733, 736, 782, 786, 787, 820– 822, 828, 833, 834, 960, 982, 988, 990, 1036, 1041, 1042, 1115, 1138, 1143, 1146, 1192, 1197	<code>\glsentrytext</code>	119, 120, 147, 173, 173, 365, 371, 452, 454, 471, 482, 563, 566, 570, 703, 705, 723, 734, 817, 819, 823, 959, 976, 988, 1114, 1132, 1144
<code>\glsentryshort</code>	121, 122, 162, 163, 172, 177, 178, 179, 262– 273, 367–369, 377–386, 455, 456, 477, 482, 484, 485, 528– 536, 567–569, 575–582, 706, 707, 728, 729, 733, 736, 779– 788, 820–822, 828–835, 960, 961, 982, 983, 987, 988, 990, 991, 1034–1043, 1115, 1116, 1137, 1138, 1143, 1146, 1190–1199		
<code>\glsentryshortaccess</code>	357, 361, 558, 560, 811, 813		
<code>\Glsentryshortpl</code>	120, 164, 178, 178, 265, 271, 272, 366, 379, 384, 385, 455, 478, 484, 530, 535, 567, 576, 580,		

173, 180, 216, 365, 371, 452–
 455, 471, 482, 486, 507, 563,
 564, 566, 570, 703–705, 722,
 723, 734, 737, 759, 816, 817,
 819, 823, 959, 976, 988, 992,
 1015, 1114, 1131, 1144, 1148, 1170
 \glstentrytextaccess
 356, 360, 557, 560, 810, 813
 \glstentrytitlecase
 175, 483, 735, 989, 1145
 \glstentrytype 100, 175, 441,
 483, 692, 735, 948, 989, 1102, 1145
 \Glsentryuseri 156, 176,
 176, 374, 474, 484, 572, 726,
 735, 825, 979, 989, 990, 1135, 1145
 \glstentryuseri 156, 176,
 374, 474, 483, 572, 726, 735,
 825, 979, 980, 989, 990, 1135, 1145
 \glstentryuseriaccess
 357, 362, 558, 561, 811, 814
 \Glsentryuserii 157,
 176, 176, 374, 475, 484, 573,
 726, 735, 826, 980, 990, 1135, 1145
 \glstentryuserii 157, 176, 374,
 475, 484, 572, 573, 726, 735,
 826, 980, 990, 1135, 1136, 1145
 \glstentryuseriiaccess
 357, 362, 558, 561, 811, 814
 \Glsentryuseriii 158,
 176, 177, 375, 475, 484, 573,
 727, 735, 826, 980, 990, 1136, 1145
 \glstentryuseriii 158,
 176, 375, 475, 484, 573, 726,
 727, 735, 826, 980, 990, 1136, 1145
 \glstentryuseriiiaccess
 358, 362, 558, 561, 811, 814
 \Glsentryuseriv 159,
 177, 177, 375, 475, 484, 573,
 727, 735, 826, 981, 990, 1136, 1146
 \glstentryuseriv 159,
 177, 375, 475, 476, 484, 573,
 727, 735, 826, 981, 990, 1136, 1145
 \glstentryuserivaccess
 358, 362, 558, 561, 811, 814
 \Glsentryuserv 160,
 177, 177, 375, 476, 484, 573,
 727, 736, 826, 981, 990, 1137, 1146
 \glstentryuserv
 160, 177, 375, 476, 484, 573,
 727, 736, 826, 981, 990, 1137, 1146
 \glstentryuservaccess
 358, 363, 558, 561, 811, 814
 \Glsentryuservi 161,
 177, 177, 376, 476, 484, 573,
 728, 736, 826, 981, 990, 1137, 1146
 \glstentryuservi 161, 177, 376, 476,
 484, 573, 574, 727, 728, 736,
 826, 827, 981, 982, 990, 1137, 1146
 \glstentryuserviaccess
 358, 363, 558, 561, 811, 814
 \glscsclocationsfalse 12, 201,
 499, 751, 897, 1006, 1052, 1162
 \glscsclocationstrue 393, 643
 \glsexpandfields
 86, 433, 683, 939, 1094
 \glsfieldaccsupp
 358, 359, 360, 558, 559, 812
 \glsfielddef 95, 438, 689, 945, 1100
 \glsfieldedef 95, 438, 688, 944, 1099
 \glsfieldfetch 96, 175, 439,
 483, 689, 735, 945, 989, 1100, 1145
 \glsfieldgdef 95, 438, 689, 945, 1099
 \glsfieldxdef 94, 438, 688, 944, 1099
 \glsfindwidesttoplevelname
 337, 637, 890
 \GLSfirst 148, 471, 723, 976, 1132
 \Glsfirst 148, 148, 471, 723, 976, 1132
 \glsfirst 148, 148, 471, 723, 976, 1132
 \glsfirstaccessdisplay
 360, 365, 371, 560, 563, 564,
 566, 571, 813, 816, 817, 819, 824
 \GLSfirstplural
 150, 472, 724, 977, 1133
 \Glsfirstplural
 150, 150, 472, 724, 977, 1133
 \glsfirstplural
 149, 150, 472, 723, 977, 1132, 1133
 \glsfirstpluralaccessdisplay
 361, 364, 372, 560, 562, 563,
 565, 571, 813, 815, 816, 818, 824
 \glsfirstpluralaccessdisplay
 364, 566, 819
 \glsngenacfmt
 120, 263–265, 270, 366, 377,
 378, 383, 455, 528–530, 534,
 567, 575, 580, 706, 780, 781,
 786, 820, 828, 833, 960, 1035,
 1036, 1041, 1115, 1191, 1192, 1196
 \glsgenentryfmt
 118, 118, 263–265, 269, 270,

275, [363](#), [377](#), [378](#), [382](#), [383](#),
 453, [528–530](#), [533](#), [534](#), [537](#),
 538, [540](#), [542–544](#), [546](#), [548](#),
 550, [565](#), [575](#), [578](#), [580](#), [704](#),
 780, [781](#), [785](#), [786](#), [789](#), [790](#),
 792, [794](#), [796](#), [798](#), [799](#), [801](#),
 803, [818](#), [828](#), [831](#), [833](#), [958](#),
 1035, [1036](#), [1040](#), [1041](#), [1043](#),
 1113, [1191](#), [1192](#), [1195](#), [1196](#), [1199](#)
 \glsgetgrouplabel [250](#), [522](#), [774](#), [1028](#), [1184](#)
 \glsgetgrouptitle [249](#), [288](#),
 292–294, [313–318](#), [334–337](#),
 340, [341](#), [521](#), [610](#), [611](#), [613](#),
 623–626, [635–637](#), [639](#), [773](#),
 864, [866](#), [867](#), [877–880](#), [889](#),
 890, [892](#), [893](#), [1028](#), [1047](#), [1183](#)
 \gls glossarymark
 . [55](#), [55](#), [415](#), [665](#), [920](#), [921](#), [1075](#)
 \gls groupheading [188](#),
 235, [238](#), [248](#), [289](#), [292–294](#),
 296, [297](#), [299](#), [307](#), [309](#), [310](#),
 313–320, [322](#), [326](#), [328](#), [330](#),
 333–338, [340](#), [341](#), [490](#), [516](#),
 521, [592](#), [611–616](#), [620–629](#),
 631–639, [742](#), [767](#), [773](#), [846](#),
 864, [866–869](#), [873–875](#), [877–](#)
 882, [884–886](#), [888–892](#), [996](#),
 1023, [1028](#), [1152](#), [1178](#), [1183](#)
 \gls groupskip
 [187](#), [188](#), [235](#), [238](#), [248](#),
 290, [293](#), [296](#), [298–300](#), [302–](#)
 304, [308](#), [309](#), [311](#), [319](#), [321](#),
 323, [327](#), [328](#), [330](#), [334](#), [335](#),
 337, [340](#), [490](#), [516](#), [521](#), [592](#),
 612, [614–618](#), [620–622](#), [627–](#)
 629, [631–633](#), [635–637](#), [639](#),
 741, [742](#), [767](#), [773](#), [845](#), [865](#),
 866, [868–872](#), [874–876](#), [880–](#)
 882, [885–890](#), [892](#), [996](#), [1023](#),
 1028, [1151](#), [1152](#), [1178](#), [1183](#)
 \glshyperfirstfalse [271](#),
[383](#), [534](#), [580](#), [786](#), [833](#), [1041](#), [1197](#)
 \glshyperfirsttrue
 [32](#), [405](#), [654](#), [909](#), [1064](#)
 \glshyperlink . [180](#), [216](#), [486](#), [507](#),
[737](#), [759](#), [992](#), [1015](#), [1148](#), [1170](#)
 \glshypernavsep
 288, [288](#), [610](#), [611](#), [863](#), [864](#), [1047](#)
 \glshypernumber [53](#), [252](#), [254](#),
 414, [523](#), [524](#), [664](#), [775](#), [776](#),
 920, [1029](#), [1031](#), [1075](#), [1185](#), [1186](#)
 \glsifhyper [125](#), [458](#), [709](#)
 \glsifhyperon
[127](#), [458](#), [459](#), [709](#), [710](#), [963](#), [1118](#)
 \glsIfListOfAcronyms
 [21](#), [21](#), [22](#), [398](#), [648](#), [902](#), [1057](#)
 \glsifmeasuring
 [107](#), [107](#), [108](#), [953](#), [1107](#), [1108](#)
 \glsifplural [118](#), [120](#),
[121](#), [141–146](#), [162–169](#), [268](#),
[363](#), [366](#), [367](#), [369–371](#), [381](#),
[451](#), [454–456](#), [468–470](#), [476–](#)
[481](#), [533](#), [538](#), [542](#), [544](#), [546](#),
[562](#), [565](#), [567–570](#), [577](#), [701](#),
[704](#), [706](#), [719–722](#), [728–732](#),
[784](#), [791](#), [794](#), [796](#), [799](#), [815](#),
[818](#), [820–823](#), [830](#), [958](#), [960](#),
[973–975](#), [982–986](#), [1039](#), [1113](#),
[1115](#), [1129–1131](#), [1137–1142](#), [1195](#)
 \glsifusedtranslatordict
 [30](#), [404](#), [653](#), [908](#), [1062](#)
 \glsifusetranslator
 [30–32](#), [50](#), [341](#), [342](#), [403–](#)
[405](#), [412](#), [413](#), [588](#), [604](#), [653](#),
[654](#), [662](#), [663](#), [841](#), [857](#), [907–](#)
[909](#), [918](#), [1062–1064](#), [1072](#), [1073](#)
 \glsignore [182](#), [486](#), [738](#), [993](#), [1148](#)
 \glsindexingsetting
 [15](#), [18](#), [198](#), [200](#), [899–901](#),
[1004](#), [1006](#), [1053](#), [1055](#), [1160](#), [1161](#)
 \glsindexonlyfirstfalse
 [32](#), [405](#), [654](#), [909](#), [1064](#)
 \glsinlinedescformat [290](#),
[291](#), [595](#), [611](#), [612](#), [848](#), [864](#), [865](#)
 \glsinlinedopostchild [289](#),
[290](#), [594](#), [611](#), [612](#), [847](#), [864](#), [865](#)
 \glsinlineemptydescformat
 [289](#), [290](#),
[291](#), [594](#), [611](#), [612](#), [848](#), [864](#), [865](#)
 \glsinlineifhaschildren [290](#), [290](#)
 \glsinlinenameformat [289](#),
[291](#), [594](#), [611](#), [612](#), [848](#), [864](#), [865](#)
 \glsinlineparentchildseparator
[290](#), [290](#), [595](#), [611](#), [612](#), [848](#), [865](#)
 \glsinlinepostchild
[290](#), [291](#), [595](#), [611](#), [612](#), [848](#), [865](#)
 \glsinlineseparator
[290](#), [290](#), [595](#), [611](#), [612](#), [848](#), [865](#)

<code>\glsinlinesubdescformat</code>	<code>\glslink</code> . . 126, 126 , 261, 269–271,
. . . . 290, 291 , 595, 612, 848, 865	275, 382, 384, 458, 527, 533–
<code>\glsinlinesubnameformat</code>	535, 538, 578–581, 709, 779,
. . . . 290, 291 , 595, 612, 848, 865	785, 787, 790, 831–834, 962,
<code>\glsinlinesubseparator</code>	963, 1034, 1040–1043, 1117,
. . . . 290, 290 , 595, 612, 848, 865	1118, 1189, 1190, 1196, 1197, 1199
<code>\glsinsert</code> . . . 118–122, 141–146,	<code>\glslinkcheckfirsthyperhook</code> . . .
162–169, 268, 269, 363–371,	. . . 127, 127 , 459, 710, 963, 1118
381, 382, 451–456, 468–470,	<code>\glslinkpostsetkeys</code>
476–481, 533, 539, 542–544,	127 , 128, 459, 710, 963, 964, 1118
546, 562–570, 578, 702–707,	<code>\glslinkvar</code>
719–722, 728–732, 784, 785,	. . . 125, 125 , 458, 709, 962, 1117
791, 794, 796, 798, 799, 815–	<code>\glslistdottedwidth</code>
823, 831, 959–961, 973–975, 295, 295 , 596, 613, 849, 867
982–986, 1039, 1040, 1113–	<code>\glslistexpandedname</code> . . 292, 292 , 866
1116, 1129–1131, 1137–1142, 1195	<code>\glslistgroupheaderfmt</code>
<code>\glskeylisttok</code> 260 , 261,	291 , 292–294, 612, 613, 865–867
387, 526, 527, 537–545, 547–	<code>\glslistinit</code> 292, 292 , 865, 866
550, 582–586, 778, 779, 789–	<code>\glslistnavigationitem</code>
803, 835–839, 1033, 1034, 1189	292 , 293, 294, 612, 613, 865–867
<code>\glslabel</code> 99, 118–	<code>\glslocalreset</code> . . . 109 , 111, 446,
122, 126–128, 162–169, 263–	447, 697, 698, 954, 955, 1109, 1110
265, 268–270, 363–368, 377,	<code>\glslocalresetall</code>
378, 381–383, 440, 451–456, 111 , 447, 698, 955, 1110
458, 459, 477–481, 528–530,	<code>\glslocalunset</code> 110 , 111, 124, 447,
533, 534, 538, 539, 541–544,	468–471, 697, 698, 719–722,
546, 562–568, 575, 577, 578,	954, 955, 962, 1109, 1110, 1117
580, 691, 702–707, 709, 710,	<code>\glslocalunsetall</code>
728–732, 780, 781, 784–786, 111 , 447, 698, 955, 1110
790, 791, 794, 796, 798, 799,	<code>\glslocationcstoencap</code> . 53 , 920, 1075
815–821, 828, 830, 831, 833,	<code>\glslongaccessdisplay</code>
947, 958–961, 963, 964, 982– 362 , 368, 370,
986, 1035, 1036, 1039–1041,	371, 377–386, 561, 568, 570,
1102, 1113–1116, 1118, 1138–	575–583, 814, 821, 823, 828–836
1142, 1191, 1192, 1195, 1196	<code>\glslongaccesskey</code> 387 , 587, 840
<code>\glslabelhypertarget</code> 138 , 1127	<code>\glslongkey</code> 256 ,
<code>\glslabelhypertargetdefs</code>	387, 525, 587, 777, 840, 1031, 1187
. 138 , 139 , 1127 , 1128	<code>\glslongpluralaccessdisplay</code> . . .
<code>\glslabelhypertargetprefix</code> 362 , 368,
. 138 , 139 , 1127 , 1128	378, 379, 381–385, 561, 568,
<code>\glslabelhypertargetvalue</code>	575, 576, 578–581, 583, 814,
. 138 , 139 , 1127	821, 822, 828, 829, 831–834, 836
<code>\glslabeltok</code>	<code>\glslongpluralaccesskey</code> 387 , 587, 840
. . . . 260 , 261, 526, 527, 537–	<code>\glslongpluralkey</code> 256 ,
550, 582–586, 778, 779, 789–	387, 525, 587, 777, 840, 1031, 1187
794, 796–803, 835–839, 1033, 1189	<code>\glslongtok</code> . . . 260 , 261–265, 269,
<code>\glsletentryfield</code>	270, 377, 378, 382, 383, 387,
. 170 , 481, 732, 986, 1142	526–530, 533, 534, 537–550,
<code>\Glslink</code> . . . 126, 126 , 963, 1117, 1118	575, 578, 580, 582–586, 778–
	781, 785, 786, 789–804, 828,

831, 833, 835–839, 1033–1036,
 1040, 1041, 1189–1192, 1195, 1197
 \glslowercase 4, 893, 1048
 \glsLTpenaltycheck
 305, 306, 619, 872, 873
 \glsmakefirsttuc 42, 915, 1070
 \glsmcols 313,
 313, 314–318, 623–626, 876–880
 \glsmeasuredepth 108, 953, 1108
 \glsmeasureheight
 .. 108, 138, 953, 972, 1108, 1127
 \glsmeasurewidth
 109, 337–339, 953, 1108
 \glsmfuaddmap 5, 6,
 94, 115, 116, 126, 142, 144,
 146–162, 164, 166, 168, 246,
 247, 258, 259, 344, 345, 347,
 348, 893, 944, 957, 958, 963,
 973–984, 986–991, 1027, 1028,
 1032, 1033, 1048, 1099, 1112,
 1113, 1118, 1129–1141, 1183, 1188
 \glsmfublocker 5, 6, 94, 142,
 145, 147–161, 163, 165, 167,
 169, 893, 944, 974–982, 984–
 986, 1048, 1099, 1129–1140, 1142
 \glsmfuexcl . 5, 5, 6, 171, 173–179,
 181, 893, 992, 1048, 1143–1148
 \glsmoveentry 105, 444, 695, 951, 1106
 \GLSname 151, 473, 724, 977, 978, 1133
 \Glsname 151, 151, 472, 724, 977, 1133
 \glsname 150, 151, 472, 724, 977, 1133
 \glsnameaccessdisplay
 360, 372, 376,
 377, 559, 571, 574, 812, 824, 827
 \glsnamefont 246,
 251, 376, 520, 521, 523, 553,
 554, 574, 772, 773, 775, 806,
 807, 827, 1027, 1029, 1183, 1185
 \glsnavhypergroupdotarget . 287, 287
 \glsnavhyperlink 285,
 288, 609–611, 863, 864, 1046, 1047
 \glsnavhyperlinkname
 285, 285, 287, 609, 610, 863, 1046
 \glsnavhypertarget
 . 287, 292–294, 313–318, 334,
 336, 337, 341, 609, 613, 623–
 626, 635–637, 639, 863, 866,
 867, 877–880, 889, 890, 893, 1046
 \glsnavigation
 . 288, 292–294, 313–318, 334,
 335, 337, 341, 610, 613, 623–
 626, 635–637, 639, 863, 866,
 867, 877–880, 889, 890, 892, 1047
 \glsnavigationitem 288, 288
 \glsnextpages 12, 82,
 219, 244, 393, 430, 509, 518,
 643, 680, 761, 770, 897, 936,
 1016, 1026, 1052, 1091, 1172, 1181
 \glsnoexpandfields
 87, 433, 683, 939, 1094
 \glsnogroupskipfalse
 14, 394, 644, 898, 1052
 \glsnoidxdisplayloc 203, 204, 239,
 501, 502, 516, 752, 753, 768,
 1008, 1009, 1024, 1163, 1164, 1179
 \glsnoidxdisplaylocclishandler
 202, 239, 500, 516,
 752, 768, 1007, 1023, 1163, 1179
 \glsnoidxloclist
 202, 234, 236, 238, 239, 239,
 500, 515, 516, 752, 767, 768,
 1007, 1022, 1023, 1162, 1178, 1179
 \glsnoidxloclisthandler
 .. 239, 239, 516, 768, 1023, 1179
 \glsnoidxnumberlistloophandler
 203,
 204, 501, 753, 1008, 1163, 1164
 \glsnoidxprecontenthook .. 233, 233
 \glsnoidxprenumberlist
 82, 238, 239,
 936, 1022, 1023, 1091, 1178, 1179
 \glsnoidxstripaccents 26, 26, 401,
 402, 650, 651, 905, 906, 1059, 1060
 \glsnomakeindexwarning
 58, 190, 417,
 492, 667, 743, 922, 998, 1077, 1154
 \glsnonextpages ... 82, 219, 243,
 430, 509, 518, 680, 761, 770,
 936, 1016, 1026, 1090, 1172, 1181
 \glsnopostdotfalse
 13, 394, 644, 898, 1052
 \glsnoxindywarning 53, 58,
 59–61, 64–66, 183, 184, 414,
 416–419, 421, 422, 487, 488,
 664, 667–669, 671, 672, 738,
 739, 920, 922–928, 993, 994,
 1074, 1077–1079, 1081–1083, 1149
 \glsnumberformat
 53, 180, 414, 485, 664,
 737, 920, 991, 1074, 1075, 1147

<code>\glsnumberlistloop</code>	203, 203 , 501, 752 , 753 , 1008, 1163 , 1164	<code>\glspl</code>	116, 143 , 144, 274, 450, 469, 537, 701, 721, 788, 958, 974, 1043, 1113, 1130, 1199
<code>\glsnumbersgroupname</code>	41, 48 , 51, 250, 409, 412, 413, 522, 659 , 662, 663, 774, 915, 917, 919, 1028, 1069, 1072, 1073, 1184	<code>\GLSplural</code> . . .	149 , 472, 723, 977, 1132
<code>\glsnumlistlastsep</code>	180, 180 , 202, 486, 500, 737, 752, 992, 1007, 1147, 1148, 1163	<code>\Glsplural</code> 149 , 149 , 472, 723, 977, 1132	
<code>\glsnumlistparser</code>	180, 199, 486, 498, 737, 750, 992, 1005, 1147, 1161	<code>\glsplural</code>	148 , 149, 472, 723, 976, 977, 1132
<code>\glsnumlistsep</code>	180, 180 , 202, 486, 500, 737, 752, 992, 1007, 1147, 1148, 1163	<code>\glspluralaccessdisplay</code> . .	360 , 363, 364, 372, 560, 562, 563, 565, 571, 813, 815, 816, 818, 824
<code>\glsopenbrace</code> . . .	63, 64, 183 , 187, 188, 420, 487, 490, 592, 670, 738, 741, 742, 845, 846, 926, 993, 996, 1081, 1148, 1151, 1152	<code>\glspluralsuffix</code> 48, 48 , 102, 261, 264, 265, 378, 379, 383–385, 412, 442, 443, 529, 530, 575, 576, 579– 581, 662, 693, 781, 782, 828, 829, 832–835, 917, 949, 1034– 1037, 1072, 1104, 1189, 1191, 1192
<code>\glsorder</code> 33, 33 , 195–198, 231, 405, 406, 495–498, 513, 655, 747– 750, 764, 765, 910, 1002, 1003, 1005, 1020, 1064, 1157–1160, 1176		<code>\glspostdescription</code> 13 , 51, 291, 293, 294, 296, 307, 319, 326, 333– 336, 339, 340, 394, 413, 595– 603, 612–614, 620, 627, 631, 635–638, 644, 663, 848–856, 865, 866, 868, 873, 874, 880, 884, 888–892, 898, 919, 1052, 1074
<code>\glsorg@endtheglossary</code>	6, 6 , 390, 640, 894, 1048	<code>\glspostinline</code> 289, 291 , 611, 612, 864, 865
<code>\glsorg@PrintChanges</code>	6, 390, 640, 894, 1048	<code>\glspostlinkhook</code> .	126, 126 , 141– 146, 162–169, 369–371, 458, 468–471, 477–481, 569, 570, 709, 720–722, 728–732, 822, 823, 963, 973–976, 982–986, 1117, 1118, 1129–1131, 1138–1142
<code>\glsorg@theglossary</code>	6, 6 , 390, 640, 894, 1048	<code>\glsprefixsep</code>	346, 346 , 347–350, 606–609, 859–862
<code>\glspagelistwidth</code>	296 , 297, 298, 301, 304, 305, 307 , 309–312, 318 , 320–322, 324, 325, 326 , 328– 331, 614, 615, 617–623, 627, 628, 630–634, 867–876, 880–887	<code>\glsprestandardsort</code> 15 , 16, 395, 645, 899, 1054
<code>\glspar</code>	51 , 413, 663, 919, 1074	<code>\glsquote</code> 45, 183 , 487, 738, 993, 1148	
<code>\glspatchLToutput</code>	302–305, 306 , 617–619, 871, 872	<code>\glsrefentry</code> 244 , 519, 771, 1026, 1182	
<code>\glspatchtabularx</code>	109 , 446, 697, 954, 1108	<code>\glsreset</code>	109 , 111, 446, 447, 697, 698, 954, 955, 1109, 1110
<code>\glspenaltygroupskip</code>	302–304, 305 , 617–619, 871, 872	<code>\glsresetall</code> .	111 , 447, 698, 955, 1110
<code>\glspercentchar</code> .	89, 90, 183 , 187, 189, 434, 487, 489, 491, 685, 738, 741, 742, 941, 993, 996, 997, 1095, 1096, 1148, 1151, 1152	<code>\glsresetcurrcountfalse</code>	112, 955, 1110
<code>\GLSpl</code>	145 , 470, 721, 975, 1130	<code>\glsresetentrycounter</code> 244 , 519, 770, 1026, 1181
<code>\Glspl</code>	117, 144, 144 , 274, 450, 469, 537, 701, 721, 788, 958, 974, 1043, 1113, 1130, 1199	<code>\glsresetentrylist</code>	187, 233, 236, 243, 243 , 490, 514, 518, 592, 741, 766, 770, 845, 996, 1021, 1025, 1151, 1177, 1181

<code>\glsresetsubentrycounter</code>	917, 918, 1016, 1072, 1073, 1172
. 244 , 245 , 290 ,	
518 , 519 , 595 , 611 , 770 , 771 ,	
848 , 864 , 1026 , 1027 , 1181 , 1182	
<code>\glsrestoreLToutput</code> . . . 305 , 619 , 872	
<code>\glssanitizeortfalse</code>	
. 29 , 403 , 652 , 906 , 907 , 1061	
<code>\glssanitizeorttrue</code>	
. 29 , 402 , 403 , 652 , 906 , 907 , 1061	
<code>\gls savenumberlistfalse</code>	
. 12 , 393 , 643 , 897 , 1052	
<code>\gls savewritesfalse</code>	
. 39 , 408 , 658 , 913 , 1068	
<code>\glssee</code> 215 , 507 , 759 , 1014 , 1170	
<code>\glsseeformat</code> . . . 186 , 201 , 203 , 215 ,	
489 , 499 , 501 , 507 , 591 , 740 ,	
751–753 , 759 , 845 , 995 , 1006 ,	
1008 , 1014 , 1151 , 1162–1164 , 1170	
<code>\glsseeitem</code> 215 ,	
216 , 507 , 759 , 1014 , 1015 , 1170	
<code>\glsseeitemformat</code>	
. 216 , 216 , 507 , 759 , 1015 , 1170	
<code>\glsseeelastsep</code> 215 ,	
216 , 507 , 759 , 1014 , 1015 , 1170	
<code>\glsseelist</code>	
. 215 , 215 , 507 , 759 , 1014 , 1170	
<code>\glsseesep</code>	
. 215 , 216 , 507 , 759 , 1015 , 1170	
<code>\gls sentencecase</code> 4 ,	
122 , 123 , 126 , 146 , 170–172 ,	
893 , 961 , 963 , 975 , 987 , 988 ,	
1048 , 1116 , 1117 , 1131 , 1142 , 1143	
<code>\glsSetAlphaCompositor</code>	
. 53 , 414 , 664 , 919 , 920 , 1074	
<code>\glsSetCompositor</code>	
. 52 , 414 , 664 , 919 , 1074	
<code>\glssetexpandfield</code> 24 , 24 ,	
28 , 29 , 400 , 402 , 403 , 649–653 ,	
904 , 906 , 907 , 1058 , 1061 , 1062	
<code>\glssetnoexpandfield</code> 24 ,	
24 , 25 , 28 , 29 , 400 , 402 , 403 ,	
649–652 , 904 , 906 , 907 , 1058–1061	
<code>\GlsSetQuote</code> 99 ,	
189 , 190 , 440 , 490 , 492 , 494 ,	
691 , 742–744 , 746 , 947 , 997 ,	
998 , 1001 , 1102 , 1152 , 1154 , 1157	
<code>\glsSetSuffixF</code> 53 , 414 , 664 , 920 , 1074	
<code>\glsSetSuffixFF</code> 53 , 414 , 664 , 920 , 1074	
<code>\glssettoctitle</code> 48 , 50 , 218 ,	
411 , 413 , 509 , 661 , 663 , 760 ,	
917 , 918 , 1016 , 1072 , 1073 , 1172	
<code>\glssetwidest</code> 338 , 637 , 891	
<code>\GlsSetWriteIstHook</code>	
. 190 , 492 , 743 , 998 , 1154	
<code>\GlsSetXdyCodePage</code>	
. 66 , 422 , 672 , 928 , 1082 , 1083	
<code>\GlsSetXdyFirstLetterAfterDigits</code>	
. 183 , 487 , 738 , 993 , 1149	
<code>\GlsSetXdyLanguage</code>	
. 66 , 422 , 672 , 927 , 928 , 1082	
<code>\GlsSetXdyLocationClassOrder</code> . .	
. 64 , 421 , 671 , 927 , 1081	
<code>\GlsSetXdyMinRangeLength</code>	
. 184 , 487 , 488 , 739 , 994 , 1149	
<code>\GlsSetXdyNumberGroupOrder</code>	
. 184 , 487 , 739 , 993 , 994 , 1149	
<code>\GlsSetXdyStyles</code>	
. 65 , 422 , 672 , 927 , 1082	
<code>\glsshortaccessdisplay</code> 172 ,	
361 , 367–369 , 377–386 , 482 ,	
560 , 567–569 , 575–583 , 733 ,	
813 , 820–822 , 828–836 , 988 , 1143	
<code>\glsshortaccesskey</code> 387 , 587 , 840	
<code>\glsshortaccsupp</code> 359 , 359 , 559 , 812	
<code>\glsshortkey</code> 256 ,	
387 , 525 , 587 , 777 , 840 , 1031 , 1187	
<code>\glsshortplaccsupp</code> 359 , 559 , 812	
<code>\glsshortpluralaccessdisplay</code> . .	
. 362 , 366 , 368 , 378 ,	
379 , 382–385 , 560 , 567 , 568 ,	
575 , 576 , 579–581 , 583 , 813 ,	
820–822 , 828 , 829 , 832–834 , 836	
<code>\glsshortpluralaccesskey</code>	
. 387 , 587 , 840	
<code>\glsshortpluralkey</code> 256 ,	
387 , 525 , 587 , 777 , 840 , 1031 , 1187	
<code>\glsshorttok</code>	
. 260 , 261 , 387 , 526 , 527 , 537–	
550 , 582–586 , 778 , 779 , 789–	
803 , 835–839 , 1033 , 1034 , 1189	
<code>\glsshowaccsupp</code> 7 ,	
9 , 391 , 392 , 641 , 895 , 1049 , 1050	
<code>\glsshowtarget</code> 8 ,	
8 , 391 , 392 , 641 , 895 , 1049 , 1050	
<code>\glsshowtargetfont</code>	
. 8 , 8 , 391 , 641 , 895 , 1050	
<code>\glsshowtargetfonttext</code> 8 , 8 , 895 , 1050	
<code>\glsshowtargetinner</code>	
. 8 , 8 , 895 , 1049 , 1050	

<code>\glsshowtargetouter</code>	520, 521, 595–604, 612–616,
..... 8, 8 , 391, 641, 895, 1050	620–622, 627–629, 631–633,
<code>\glsshowtargetsymbol</code>	635–638, 772, 773, 848–857,
..... 8, 8 , 391, 641, 895, 1050	865–869, 873–875, 880–882,
<code>\glssortnumberfmt</code> 16 , 17 , 18 , 395,	884–892, 953, 1027, 1108, 1183
396, 645, 646, 899, 900, 1054, 1055	<code>\glstexorpdfstring</code>
<code>\glsspace</code>	170, 170 , 171, 175, 209, 987,
257, 257 , 525, 777, 790, 1032, 1187	989, 1011, 1142, 1143, 1145, 1167
<code>\glstepentry</code> 244 , 245, 519,	<code>\GLStext</code> 147 , 471, 722, 976, 1131
770, 771, 1026, 1027, 1181, 1182	<code>\Glstext</code>
<code>\glstepsubentry</code>	147, 147 , 471, 723, 976, 1131, 1132
244,	<code>\glstext</code>
245, 519, 771, 1026, 1027, 1182	147, 147 , 471, 722, 976, 1131, 1132
<code>\glssubentrycounterfalse</code>	<code>\glstextaccessdisplay</code>
..... 15, 395, 645, 899, 1053	360, 365, 371, 560, 563, 564,
<code>\glssubentrycounterlabel</code> .. 245,	566, 570, 813, 816, 817, 819, 823
245 , 519, 771, 1026, 1027, 1182	<code>\glstextformat</code> ... 118 , 126, 128,
<code>\glssubentryitem</code> . 245 , 290, 293–	451, 458, 459, 701, 709, 710,
296, 298, 299, 307, 309, 311,	958, 962, 964, 1113, 1117, 1118
319, 321, 322, 326, 328, 330,	<code>\glstextup</code>
333, 335, 336, 339, 519, 595–	48,
600, 602–604, 612–616, 620–	256 , 385, 412, 525, 581, 662,
622, 627–629, 631–633, 635,	777, 834, 917, 1031, 1072, 1187
636, 638, 771, 848–853, 855–	<code>\glstildechar</code>
857, 865–869, 874, 875, 880–	60,
882, 884–886, 888–891, 1027, 1182	183 , 187, 188, 418, 487, 489,
<code>\GLSsymbol</code> .. 154 , 474, 725, 979, 1134	490, 668, 738, 741, 742, 923,
<code>\Glssymbol</code> 154 , 154 , 474, 725, 979, 1134	993, 996, 1078, 1148, 1151, 1152
<code>\glssymbol</code>	<code>\glstranslatefalse</code>
..... 153 , 154, 473, 725, 978, 979, 1134	30, 31,
<code>\glssymbolaccessdisplay</code> .. 361 ,	404, 405, 653, 654, 908, 1062, 1063
373, 377, 560, 563–565, 572,	<code>\glstranslatetrue</code> .. 31, 32, 404,
574, 813, 816–818, 825, 827, 828	405, 653, 654, 908, 909, 1063, 1064
<code>\glssymbolnav</code> ... 288 , 610, 864, 1047	<code>\glstreechildpredesc</code>
<code>\GLSsymbolplural</code>	333 , 334, 335, 634–636, 888, 889
..... 155 , 474, 725, 979, 1135	<code>\glstreegroupheaderfmt</code>
<code>\Glssymbolplural</code> 313–318, 332 , 334–337,
.... 155 , 155 , 474, 725, 979, 1135	340, 341, 623–626, 634–637,
<code>\glssymbolpluralaccessdisplay</code> .	639, 877–880, 888–890, 892, 893
..... 361 , 374, 560,	<code>\glstreeindent</code>
562, 563, 572, 813, 815, 816, 825	335, 336, 336 , 338–340, 599–
<code>\glssymbolsgroupname</code>	601, 636, 638, 852–854, 889–892
41,	<code>\glstreeitem</code>
48 , 51, 250, 409, 412, 413, 522,	313, 314, 332,
659, 662, 663, 774, 914, 917,	332 , 333, 623, 624, 634, 877, 888
919, 1028, 1069, 1072, 1073, 1184	<code>\glstreenamebox</code>
<code>\glstarget</code>	338 , 339, 340, 637, 638, 891, 892
108,	<code>\glstreenamefmt</code>
246 , 291–296, 298, 299, 307,	332, 333–340, 634–638, 888–892
309–311, 319, 321–323, 326,	<code>\glstreenavigationfmt</code>
328, 330, 333–336, 339, 340, 313–318, 332 , 334,
	335, 337, 341, 623–626, 634–
	637, 639, 877–880, 888–890, 892

`\glstreepredesc` [332](#),
[333](#), [335](#), [336](#), [634–636](#), [888–890](#)
`\glstreesubitem`
[313](#), [332](#), [333](#), [623](#), [634](#), [877](#), [888](#)
`\glstreesubsubitem`
[313](#), [332](#), [333](#), [623](#), [634](#), [877](#), [888](#)
`\glstype` [126](#), [127](#), [141–](#)
[146](#), [162–169](#), [369–371](#), [459](#),
[468–470](#), [477–481](#), [569](#), [570](#),
[709](#), [710](#), [719–722](#), [728–732](#),
[822](#), [823](#), [963](#), [973–975](#), [982–](#)
[986](#), [1118](#), [1129–1131](#), [1138–1142](#)
`\glsumarkfalse` [14](#), [394](#), [644](#), [898](#), [1053](#)
`\glsumarktrue` [14](#), [394](#), [644](#), [898](#), [1053](#)
`\glsunexpandedfieldvalue`
. [96](#), [292](#), [689](#), [866](#), [945](#), [1100](#)
`\glsunset` [109](#),
[111](#), [113](#), [114](#), [124](#), [446–449](#),
[468–471](#), [697–700](#), [720–722](#),
[954–957](#), [962](#), [1109–1111](#), [1117](#)
`\glsunsetall` . [111](#), [447](#), [698](#), [955](#), [1110](#)
`\glsupacrluralsuffix` [48](#),
[265](#), [266](#), [272](#), [412](#), [530](#), [531](#),
[535](#), [540](#), [543](#), [545](#), [547](#), [662](#),
[782](#), [787](#), [792](#), [796](#), [798](#), [800](#),
[917](#), [1037](#), [1042](#), [1072](#), [1193](#), [1198](#)
`\glssupercase` [4](#),
[55](#), [56](#), [94](#), [119–122](#), [147–161](#),
[163](#), [165](#), [167](#), [169](#), [261](#), [269–](#)
[271](#), [349](#), [350](#), [364–369](#), [371–](#)
[376](#), [381](#), [382](#), [384](#), [385](#), [893](#),
[921](#), [944](#), [959–961](#), [976–986](#),
[1034](#), [1039–1042](#), [1048](#), [1075](#),
[1076](#), [1099](#), [1114–1116](#), [1131–](#)
[1139](#), [1141](#), [1142](#), [1190](#), [1195–1197](#)
`\GlsUseAcrEntryDispStyle` . . [262](#),
[263](#), [265–268](#), [270](#), [272](#), [273](#),
[379](#), [380](#), [383](#), [385](#), [386](#), [528](#),
[530–532](#), [534–536](#), [576](#), [577](#),
[579](#), [581](#), [582](#), [780](#), [782–784](#),
[786–788](#), [829](#), [830](#), [832](#), [834](#),
[835](#), [1034](#), [1035](#), [1037–1039](#),
[1041–1043](#), [1190–1194](#), [1196](#), [1198](#)
`\GlsUseAcrStyleDefs` [262](#), [263](#), [265–](#)
[268](#), [270](#), [272](#), [273](#), [379](#), [380](#),
[383](#), [385](#), [386](#), [528](#), [530–532](#),
[534–536](#), [576](#), [577](#), [579](#), [581](#),
[582](#), [780](#), [782–784](#), [786–788](#),
[829](#), [830](#), [832](#), [834](#), [835](#), [1034](#),
[1035](#), [1037–1039](#), [1041–1043](#),
[1190](#), [1191](#), [1193–1196](#), [1198](#), [1199](#)
`\GLSuseri` [156](#), [474](#), [726](#), [979](#), [980](#), [1135](#)
`\Glsuseri` [156](#), [156](#), [474](#), [726](#), [979](#), [1135](#)
`\glsuseri` [155](#), [156](#), [474](#), [726](#), [979](#), [1135](#)
`\glsuseriaccessdisplay`
. [362](#), [374](#), [561](#), [572](#), [814](#), [825](#)
`\GLSuserii`
. [157](#), [475](#), [726](#), [980](#), [1135](#), [1136](#)
`\Glsuserii` [157](#), [157](#), [475](#), [726](#), [980](#), [1135](#)
`\glsuserii` [156](#), [157](#), [474](#), [726](#), [980](#), [1135](#)
`\glsuseriiaccessdisplay`
. [362](#), [374](#), [561](#), [572](#), [573](#), [814](#), [826](#)
`\GLSuseriii` [158](#), [475](#), [727](#), [980](#), [1136](#)
`\Glsuseriii`
. [158](#), [158](#), [475](#), [726](#), [980](#), [1136](#)
`\glsuseriii`
. [157](#), [158](#), [475](#), [726](#), [980](#), [1136](#)
`\glsuseriiiaccessdisplay`
. [362](#), [375](#), [561](#), [573](#), [814](#), [826](#)
`\GLSuseriv` [159](#), [475](#), [727](#), [981](#), [1136](#)
`\Glsuseriv` [159](#), [159](#), [475](#), [727](#), [981](#), [1136](#)
`\glsuseriv`
. [158](#), [159](#), [475](#), [727](#), [980](#), [981](#), [1136](#)
`\glsuserivaccessdisplay`
. [362](#), [375](#), [561](#), [573](#), [814](#), [826](#)
`\GLSuseriv` [160](#), [476](#), [727](#), [981](#), [1137](#)
`\Glsuseriv` [160](#), [160](#), [476](#), [727](#), [981](#), [1137](#)
`\glsuseriv`
. [159](#), [160](#), [476](#), [727](#), [981](#), [1136](#), [1137](#)
`\glsuserivaccessdisplay`
. [363](#), [375](#), [561](#), [573](#), [814](#), [826](#)
`\GLSuserivi` [161](#), [476](#), [728](#), [981](#), [982](#), [1137](#)
`\Glsuserivi` [161](#), [161](#), [476](#), [728](#), [981](#), [1137](#)
`\glsuserivi` [160](#), [161](#), [476](#), [727](#), [981](#), [1137](#)
`\glsuseriviaccessdisplay` [363](#),
[376](#), [561](#), [573](#), [574](#), [814](#), [826](#), [827](#)
`\GlsWarnAddProtectedPageFmt`
. [208](#), [208](#), [1011](#), [1166](#), [1167](#)
`\glswrallowprimitivemodsfalse`
. [210](#), [1012](#), [1167](#)
`\glswrallowprimitivemodstrue`
. [504](#), [756](#)
`\glswrglossdisableanchorcmds`
. [209](#), [213](#), [253](#),
[1011](#), [1013](#), [1030](#), [1167](#), [1169](#), [1186](#)
`\glswrglossdisablelocationcmds`
. [204](#),
[207](#), [207](#), [1008](#), [1010](#), [1164](#), [1166](#)

	1092, 1093, 1098–1101, 1107, 1165, 1173, 1174, 1176, 1177, 1190		197, 199, 237, 238, 260, 262, 268, 277, 363, 366, 381, 399,
<code>\ifcseempty</code> 71, 345, 424, 425, 606, 674, 675, 859, 930, 1085		407, 415, 416, 423, 425, 426, 428, 451, 453, 455, 497, 499,
<code>\ifcsequal</code>	... 71, 425, 675, 930, 1085		514, 515, 527, 528, 532, 538,
<code>\ifcsname</code> 96, 689, 946, 1100		541, 543, 545, 546, 562, 565,
<code>\ifcsstrequal</code>	. 97, 439, 690, 946, 1101		567, 577, 648, 657, 665, 666,
<code>\ifcsstring</code>	... 97, 439, 690, 946, 1100		673, 675, 676, 678, 701, 704, 706, 749, 750, 766, 767, 778, 780, 784, 790, 794, 796, 798, 815, 818, 820, 830, 903, 911, 921, 929–932, 934, 958, 960, 1004, 1005, 1022, 1023, 1033, 1034, 1039, 1044, 1057, 1066, 1076, 1083, 1085, 1086, 1088, 1089, 1113, 1115, 1160, 1161, 1177, 1178, 1189, 1190, 1195, 1199
<code>\ifcsundef</code> 10, 17, 34, 47, 50, 53, 55, 57, 68, 69, 77, 78, 81, 100, 102, 103, 112, 123, 129, 139, 140, 205, 217, 228, 240, 241, 245, 249–252, 255, 262, 360–363, 392, 396, 406, 411, 412, 414–416, 423, 427– 429, 441, 443, 448, 457, 460, 467, 495, 496, 502, 507, 509, 511, 517, 520, 522, 523, 525, 528, 559–561, 594, 610, 642, 645, 655, 661, 662, 664–666, 673, 677–679, 692–694, 699, 708, 710, 711, 719, 747, 748, 754, 759, 761, 763, 768, 769, 771, 773–776, 779, 812–814, 847, 863, 896, 899, 910, 916, 918, 920, 922, 929, 933–935, 948–950, 956, 961, 964, 972, 1009, 1015, 1018, 1024, 1027– 1029, 1031, 1034, 1047, 1051, 1054, 1064, 1071, 1073–1076, 1083, 1084, 1088–1090, 1102, 1104, 1105, 1110, 1116, 1119, 1128, 1164, 1171, 1174, 1179, 1180, 1182, 1184, 1185, 1187, 1190	<code>\ifdefequal</code> 35, 71– 74, 85, 86, 91, 101, 105, 238, 360, 406, 407, 424–426, 432, 433, 435, 441, 444, 515, 559, 656, 674–676, 682, 683, 686, 692, 695, 767, 812, 910, 911, 930–932, 938, 939, 942, 948, 951, 1023, 1065, 1085, 1086, 1092–1094, 1096, 1103, 1106, 1178
<code>\ifcsvoid</code> 72, 249, 931, 1086	<code>\ifdefstrequal</code>	97, 439, 690, 946, 1101
<code>\ifdef</code> 5, 43, 73, 83, 89, 90, 109, 124, 170, 171, 202, 203, 209, 256, 292, 332, 345, 358, 425, 426, 430, 434, 446, 458, 481, 483, 500, 501, 525, 559, 606, 634, 675, 676, 680, 684, 685, 697, 709, 733, 735, 752, 753, 777, 812, 859, 865, 888, 893, 915, 931, 936, 940, 941, 954, 962, 987, 1007, 1008, 1011, 1031, 1048, 1070, 1086, 1091, 1095, 1108, 1117, 1142, 1162, 1163, 1167, 1187	<code>\ifdefstring</code> 13, 49, 75, 195–197, 200, 230, 231, 237, 239, 351, 394, 412, 426, 427, 495–498, 512, 513, 515, 516, 554, 643, 662, 677, 747–750, 764, 765, 767, 768, 807, 897, 918, 932, 1002, 1003, 1005, 1019, 1020, 1022, 1023, 1052, 1072, 1087, 1157– 1159, 1161, 1175, 1176, 1178, 1179
<code>\ifdefempty</code> 22, 36, 56, 57, 67, 72, 73, 78, 118, 120,	<code>\ifdefvoid</code> 25, 26, 34, 49, 104, 195, 196, 238, 400, 401, 444, 515, 516, 650, 695, 767, 904, 910, 917, 951, 1002, 1003, 1022, 1023, 1059, 1065, 1072, 1105, 1157, 1158, 1178, 1179
		<code>\ifdim</code> 264, 306, 337, 530, 619, 637, 781, 873, 891, 1036, 1192
		<code>\iffalse</code> 103, 110, 444, 447, 694, 698, 950, 955, 1105, 1109
		<code>\IfFileExists</code> 12, 30, 31, 196, 197, 220, 394, 403, 404, 496, 497, 509, 643,

653, 654, 748, 749, 761, 897, 907, 908, 1003, 1004, 1017, 1052, 1062, 1063, 1158–1160, 1172	\ifglshasfielddefeq
\ifglossaryexists 54, 66, 68, 70, 194–196, 414, 422–424, 495–497, 664, 672–674, 746– 749, 920, 927, 929, 930, 1001– 1003, 1075, 1082–1084, 1157–1159 97, 439, 690, 946, 1101
\ifglssanitize@description	\ifglshasfieldvoid 72, 931, 1086
. 28, 402, 651, 906, 1060	\ifglshaschildren 70, 290, 424, 595, 611, 674, 848, 864, 930, 1084
\ifglssanitize@name	\ifglshasdesc 71, 290, 425, 611, 675, 864, 930, 1085
. 28, 402, 652, 906, 1061	\ifglshasfield 72, 425, 675, 931, 1086
\ifglssanitize@symbol	\ifglshaslong 72, 115–117, 171, 172, 263–265, 268, 270, 377, 378, 381, 383, 425, 450, 481, 528– 530, 533, 534, 544, 575, 577, 580, 675, 700, 701, 733, 780, 781, 784, 786, 796, 828, 830, 833, 931, 957, 958, 987, 1035, 1036, 1039, 1041, 1085, 1112, 1113, 1143, 1191, 1192, 1195, 1196
. 28, 402, 652, 906, 1061	\ifglshasparent 71, 223, 228, 234, 238, 337, 424, 511, 515, 637, 674, 763, 767, 891, 930, 1018, 1022, 1085, 1174, 1178
\ifglsxindy@glslnumbers	\ifglshasprefix 345, 346, 347, 349, 606, 607, 609, 859–862
. 66, 422, 672, 928, 1083	\ifglshasprefixfirst 345, 346, 347, 349, 606–609, 859–862
\ifglssacrdescription	\ifglshasprefixfirstplural 345, 347, 348, 350, 606–609, 859–862
. 32, 549, 802, 909, 1064	\ifglshasprefixplural 345, 347–349, 606–609, 859–862
\ifglssacrdua 33, 540, 545, 547, 549, 792, 798, 800, 802, 909, 1064	\ifglshasshort 72, 216, 425, 675, 931, 1015, 1086, 1170
\ifglssacrfootnote 32, 126, 459, 549, 709, 802, 909, 963, 1064, 1118	\ifglshassymbol 71, 333–336, 339, 340, 425, 538, 542, 546, 635, 636, 638, 675, 791, 794, 799, 888–892, 930, 1085
\ifglssacrone 20, 397, 647, 901, 1056	\ifglshyperfirst 126, 459, 709, 963, 1118
\ifglssacrshortcuts	\ifglslindexonlyfirst 207, 503, 755, 1010, 1166
. 46, 275, 410, 537, 660, 789, 916, 1043, 1070, 1199	\ifglslnogroupskip 293, 296, 298, 299, 302–304, 308, 309, 311, 319, 321, 323, 326, 328, 330, 334, 335, 337, 340, 612, 614–618, 620–622, 627–629, 631–633, 635–637, 639, 866, 868, 869, 871, 872, 874, 875, 880–882, 885–890, 892
\ifglssacrsmalldcaps 33, 540, 541, 543, 545, 547, 548, 792, 793, 795–798, 800, 909, 1064	\ifglslnonumberlist 243, 518, 770, 1025, 1181
\ifglssacrsmalld 33, 540, 541, 543, 545, 792, 793, 795–798, 909, 1064	
\ifglssautomake 37, 38, 43, 199, 407, 499, 657, 750, 912, 915, 1005, 1066, 1067, 1070, 1161	
\ifglssdescsuppressed 71, 289, 425, 611, 675, 864, 930, 1085	
\ifglssentrycounter 14, 15, 46, 244, 245, 394, 395, 410, 519, 644, 660, 770, 771, 898, 916, 1026, 1027, 1053, 1071, 1181, 1182	
\ifglssentryexists 69, 69, 70, 87– 89, 98, 101, 423, 424, 433, 434, 440, 442, 673, 674, 683, 684, 691, 692, 929, 939, 940, 947, 948, 1084, 1094, 1095, 1102, 1103	
\ifglssesclocations 131, 207–209, 504, 756, 965, 1010, 1011, 1120, 1166, 1167	
\ifglssfieldcseq 97, 439, 690, 946, 1101	

<code>\ifglsnopostdot</code>	13, 394, 644, 898, 1052	<code>\ifignoredglossary</code>	78,
<code>\ifglslnumberline</code>		100, 104, 207, 217, 428, 441,
	58, 416, 666, 922, 1077		444, 503, 508, 678, 692, 695,
<code>\ifglslresetcurrcount</code>		755, 759, 934, 948, 951, 1010,
	.. 112, 113, 955, 956, 1110, 1111		1015, 1089, 1103, 1105, 1165, 1171
<code>\ifglssanitizesort</code>	<code>\ifin@</code> 42, 410, 660, 915, 1070
	... 25, 28, 29, 400, 402, 403,	<code>\ifinlistcs</code>	... 232, 240, 514, 517,
	650, 652, 904, 906, 907, 1059, 1061		765, 768, 1021, 1024, 1177, 1179
<code>\ifglssavenumberlist</code>	84, 199, 216,	<code>\ifinner</code> 8, 391, 641, 895, 1049
	431, 498, 507, 681, 750, 759,	<code>\ifKV@glslink@hyper</code> 127,
	937, 1005, 1015, 1092, 1161, 1170		128, 459, 710, 963, 964, 1118, 1119
<code>\ifglssavewrites</code>	.. 39, 194, 206,	<code>\ifKV@glslink@local</code>
	408, 495, 503, 658, 746, 754,		124, 468–471, 719–722, 962, 1117
	913, 1001, 1010, 1068, 1157, 1165	<code>\ifmeasuring@</code>	107, 446, 696, 952, 1107
<code>\ifglssubentrycounter</code>	... 15, 46,	<code>\ifmmode</code>	. 8, 391, 641, 895, 1049, 1050
	244, 245, 395, 411, 518, 519,	<code>\ifnum</code>	. 16, 113, 114, 197, 237, 305,
	644, 661, 770, 771, 898, 916,		306, 335, 336, 338, 339, 396,
	1026, 1027, 1053, 1071, 1181, 1182		407, 448, 449, 497, 515, 599,
<code>\ifglstoc</code> 58, 416, 666, 922, 1077		600, 619, 636, 638, 645, 657,
<code>\ifglstranslate</code> 30, 49, 403,		699, 700, 749, 767, 852–854,
	412, 653, 662, 907, 917, 1062, 1072		872, 889–892, 899, 956, 957,
<code>\ifglsucmark</code>		1004, 1022, 1054, 1111, 1160, 1178
	55, 56, 415, 665, 921, 1075, 1076	<code>\ifstrempy</code> 9,
<code>\ifglssused</code>	69, 114, 118–120, 126,		81, 392, 594, 641, 848, 895, 1050
	182, 207, 346–349, 363–366,	<code>\ifstrequal</code> 522, 773
	423, 449, 451–455, 458, 486,	<code>\ifthenelse</code> 29,
	503, 538, 541, 543, 546, 562–		46, 57, 129, 198, 403, 411, 416,
	567, 607–609, 673, 700, 702–		460, 498, 522, 549, 610, 653,
	706, 709, 738, 755, 790, 794,		661, 666, 710, 750, 774, 802,
	796, 798, 815–820, 860–862,		863, 907, 916, 922, 964, 1005,
	929, 957–960, 963, 993, 1010,		1046, 1062, 1071, 1076, 1119, 1160
	1084, 1112–1115, 1118, 1148, 1166	<code>\IfTrackedLanguage</code>
<code>\ifglswrallowprimitivemods</code> 190, 492, 744, 998, 1154
	210, 211, 504, 505, 756, 1012, 1167	<code>\IfTrackedLanguageFileExists</code>	..
<code>\ifglsexindy</code> 15, 33, 51–53,	 50, 341, 342, 412,
	58, 59, 61, 63–66, 77, 105, 106,		588, 604, 662, 841, 857, 918, 1073
	131, 183, 184, 190, 195, 196,	<code>\iftrue</code> 104, 110, 444,
	198, 212, 214, 220, 406, 413,		447, 694, 698, 950, 954, 1105, 1109
	414, 417–422, 428, 445, 461,	<code>\ifundef</code> 14, 15,
	487, 488, 492, 495, 496, 505,		76, 88, 100, 184, 189, 198, 394,
	506, 509, 588–590, 655, 663,		395, 427, 433, 441, 488, 491,
	664, 667–672, 678, 695, 696,		498, 644, 677, 683, 692, 739,
	712, 738, 739, 743, 747, 748,		742, 749, 898, 933, 939, 948,
	757, 758, 761, 841–843, 899,		994, 997, 1004, 1005, 1053,
	910, 919, 922–928, 934, 951,		1088, 1094, 1102, 1149, 1152, 1160
	952, 966, 993, 994, 998, 1002–	<code>\ifvmode</code> 181, 486, 738, 992, 1148
	1004, 1012, 1014, 1017, 1053,	<code>\ifvoid</code> 306, 619, 873
	1064, 1074, 1077–1083, 1088,	<code>\ifx</code>	.. 14, 16, 18, 21, 36, 41, 42, 47,
	1106, 1121, 1148, 1149, 1154,		58, 60, 62, 63, 66, 67, 75, 100–
	1157, 1158, 1160, 1168, 1169, 1172		103, 106, 108, 129, 132–138,

172, 185, 188, 190–193, 205, 209, 211, 212, 214, 215, 218, 219, 250, 252, 253, 277, 278, 350, 360, 394–396, 398, 409– 411, 417–420, 422, 423, 441– 443, 445, 460, 462–467, 482, 488, 490, 492–494, 502, 504– 509, 522–524, 538, 539, 541, 543, 545, 547, 549–551, 553, 559, 590–592, 594, 598–601, 644–648, 657, 659–661, 667– 670, 672, 673, 692–694, 696, 710, 711, 713–718, 733, 739, 741, 743, 744, 746, 754, 756– 760, 774–776, 789, 790, 792, 793, 795, 797, 798, 800, 802– 804, 806, 812, 843–845, 847, 848, 851–854, 898–900, 902, 911, 915, 917, 923–926, 928, 932, 948–950, 952, 953, 964, 967–972, 987, 988, 994, 996, 998–1001, 1009, 1011–1014, 1016, 1028–1030, 1044, 1053– 1057, 1066, 1069–1071, 1077, 1078, 1080–1083, 1087, 1103– 1108, 1119, 1121–1127, 1143, 1150, 1152–1154, 1156, 1165, 1167–1172, 1184–1186, 1199, 1200	<code>\inputencodingname</code>
 34, 406, 655, 910, 1065
	<code>\InputIfFileExists</code>
 89, 434, 684, 940, 1095
	<code>\int</code> 45, 234, 240, 241
	<code>\istfile</code> 204, 502, 754
	<code>\istfilename</code> 51, 52, 185, 189, 195, 197, 198, 413, 414, 488, 491, 496–498, 590, 593, 663, 664, 739, 742, 748–750, 843, 846, 919, 994, 997, 1002, 1004, 1005, 1074, 1149, 1152, 1158–1160
	<code>\item</code> 292–295, 313, 314, 333, 334, 595, 596, 598, 612– 614, 623, 624, 634, 635, 848, 849, 851, 865–867, 877, 888, 889
	J
	<code>\jobname</code> . 36, 52, 88, 89, 185, 189, 194–197, 220, 407, 414, 433, 434, 488, 491, 495–497, 509, 590, 593, 656, 664, 684, 739, 742, 747–749, 761, 843, 846, 911, 919, 940, 994, 997, 1002– 1004, 1017, 1066, 1074, 1094, 1095, 1150, 1152, 1157–1160, 1172
	K
<code>\immediate</code> 36, 88–90, 114, 194, 205, 221, 407, 433, 434, 449, 495–497, 502, 509, 510, 656, 657, 684, 685, 700, 747–749, 754, 761, 762, 911, 912, 940, 941, 957, 1002–1004, 1009, 1017, 1066, 1094–1096, 1112, 1157, 1159, 1160, 1165, 1173	<code>kernelglossredefs</code> (option) 40
<code>\in@</code> 42, 410, 660, 915, 1070	<code>\key@ifundefined</code> 91, 92, 435, 436, 686, 687, 942, 943, 1097
<code>index</code> (option) 41	keys:
<code>index</code> (style) 333	<code>access</code> 352
<code>indexgroup</code> (style) 334	<code>counter</code> 80
<code>indexhypergroup</code> (style) 334	<code>description</code> 79
<code>\indexname</code> . . . 42, 410, 660, 915, 1070	<code>descriptionaccess</code> 352
<code>indexonlyfirst</code> (option) 32	<code>descriptionplural</code> 79
<code>\indexspace</code> 291, 293, 312, 313–318, 332, 334–337, 340, 341, 612, 623–626, 634–637, 639, 865, 866, 876–880, 887–890, 892, 893	<code>descriptionpluralaccess</code> 352
<code>inline</code> (style) 289	<code>first</code> 80
<code>\input</code> 49, 117, 412, 450, 662, 701, 917, 958, 1072, 1113	<code>firstaccess</code> 352
	<code>firstplural</code> 80
	<code>firstpluralaccess</code> 352
	<code>long</code> 84
	<code>longaccess</code> 353
	<code>longplural</code> 84
	<code>longpluralaccess</code> 353
	<code>name</code> 79
	<code>nonumberlist</code> 82
	<code>parent</code> 82
	<code>plural</code> 80

500, 501, 511, 512, 515, 520, 522, 637, 657, 674–676, 685– 687, 689, 693, 732, 733, 747, 748, 752, 762–764, 767, 772– 774, 891, 910, 911, 930–932, 941–943, 945, 949, 986, 987, 1007, 1008, 1018, 1019, 1022, 1023, 1027, 1028, 1047, 1065, 1066, 1085, 1086, 1095, 1097, 1100, 1104, 1142, 1143, 1162, 1163, 1173–1175, 1178, 1183, 1184	longborder (style) 297
list (style) 292	longheader (style) 297
\listcsadd 232 , 514 , 766 , 1021 , 1177	longheaderborder (style) 297
\listcseadd 226	\longnewglossaryentry 98 , 98 , 440 , 690 , 691 , 946 , 947 , 1101 , 1102
\listcsgadd 240 , 517 , 768 , 1024 , 1179 , 1180	longplural (key) 84
\listcsxadd 227 , 228 , 510 , 511 , 762 , 1018 , 1173 , 1174	longpluralaccess (key) 353
listdotted (style) 294	\longprovideglossaryentry 98 , 440 , 691 , 947 , 1102
\listead 231 , 514 , 765 , 1021 , 1176	longragged (style) 307
listgroup (style) 293	longragged-booktabs (style) 304
listhypergroup (style) 293	longragged3col (style) 309
\listxadd 87 , 683 , 939 , 1094	longragged3col-booktabs (style) 304
\loadglsentries 117 , 117 , 450 , 451 , 701 , 958 , 1113	longragged3colborder (style) 309
locales (option) 32	longragged3colheader (style) 310
\long 98 , 254 , 440 , 524 , 691 , 776 , 947 , 1031 , 1101 , 1186	longragged3colheaderborder (style) 310
long (key) 84	longraggedbborder (style) 308
long (style) 296	longraggedheader (style) 308
long-booktabs (style) 302	longraggedheaderborder (style) 308
long-sc-short (acrstyle) 265	\LT@end@open 306 , 619 , 872
long-sc-short-desc (acrstyle) 267 , 379	\LT@err 306 , 619 , 872
long-short (acrstyle) 263 , 377	\LT@foot 306 , 619 , 873
long-short-desc (acrstyle) 266 , 379	\LT@head 306 , 619 , 873
long-sm-short (acrstyle) 265	\LT@lastfoot 306 , 619 , 873
long-sm-short-desc (acrstyle) 267 , 379	\LT@output 305 , 306 , 619 , 872
long-sp-short (acrstyle) 264	
long-sp-short-desc (acrstyle) 266	
long3col (style) 297	
long3col-booktabs (style) 303	
long3colborder (style) 298	
long3colheader (style) 298	
long3colheaderborder (style) 299	
long4col (style) 299	
long4col-booktabs (style) 303	
long4colborder (style) 300	
long4colheader (style) 300	
long4colheaderborder (style) 300	
longaccess (key) 353	

M	
\makeatletter 89 , 220 , 434 , 509 , 684 , 761 , 940 , 1017 , 1095 , 1172	
\makebox 295 , 338 – 340 , 596 , 600 , 601 , 613 , 637 , 638 , 849 , 853 , 854 , 867 , 891 , 892	
\makefirstuc 4 , 893 , 1048	
\makeglossaries 9 , 35 , 37 , 40 , 41 , 47 , 82 , 194 , 198 , 200 – 202 , 204 , 221 , 392 , 406 – 409 , 411 , 430 , 495 , 497 – 501 , 510 , 641 , 642 , 656 – 659 , 661 , 680 , 747 , 749 – 753 , 762 , 895 , 896 , 910 – 912 , 914 , 917 , 936 , 1001 , 1004 – 1008 , 1017 , 1050 , 1051 , 1065 – 1067 , 1069 , 1072 , 1090 , 1157 , 1160 – 1162 , 1164 , 1173	
\makeglossary 39 , 39 , 40 , 41 , 408 , 409 , 658 , 659 , 913 , 914 , 1068 , 1069	
makeindex (option) 33	
\makenoidxglossaries 9 , 35 , 82 , 199 , 200 , 203 , 204 , 392 , 406 , 407 , 430 ,	

498–501, 641, 642, 656, 680,
 750–753, 895, 896, 910, 911,
 936, 1005–1008, 1050, 1051,
 1065, 1066, 1090, 1161, 1162, 1164
 \MakeTextUppercase
 5, 390, 639, 893, 1048
 \MakeUppercase 563,
 564, 569, 570, 816, 817, 822, 823
 \marginpar 8, 391, 641, 895, 1050
 \markboth 55, 415, 665, 921, 1075
 \mbox . 181, 294, 317, 318, 338, 486,
 595, 613, 625, 626, 638, 738,
 848, 866, 879, 880, 891, 992, 1148
 mcolalttree (style) 316
 mcolalttreetgroup (style) 317
 mcolalttreethypergroup (style) 317
 mcolalttreespannav (style) 317
 mcolindex (style) 313
 mcolindexgroup (style) 313
 mcolindexhypergroup (style) 313
 mcolindexspannav (style) 313
 mcoltree (style) 314
 mcoltreegroup (style) 314
 mcoltreehypergroup (style) 314
 mcoltreename (style) 315
 mcoltreenamegroup (style) 315
 mcoltreenamehypergroup (style) 316
 mcoltreenamepannav (style) 316
 mcoltreespannav (style) 315
 \memUHead 55, 415, 665, 921, 1075
 \MessageBreak 23, 40, 75, 218,
 342, 350, 399, 408, 427, 508,
 553, 588, 605, 649, 658, 677,
 760, 806, 841, 858, 903, 914,
 933, 1016, 1058, 1068, 1087, 1171
 mfirstuc (option) 42
 \mfirstucMakeUppercase
 5, 390, 415, 437, 452–456,
 471–478, 480, 481, 527, 533–
 535, 542, 546, 565–568, 570–
 574, 578–581, 609, 639, 665,
 688, 702–707, 723–732, 779,
 785, 787, 794, 799, 818–821,
 823–827, 831–834, 862, 893, 1048
 \MFUaddmap 5, 893, 1048
 \MFUblocker 5, 893, 1048
 \MFUexcl 5, 893, 1048
 \MFUsentencecase 5, 5, 42,
 170, 171, 175, 893, 915, 987,
 989, 1048, 1070, 1142, 1143, 1145
 \midrule 302, 303, 617, 618, 871
 \month 185,
 189, 488, 491, 590, 593, 739,
 742, 843, 846, 994, 997, 1150, 1152
 \msg 222

 N
 \n 189,
 491, 593, 742, 743, 846, 997, 1153
 name (key) 79
 \NeedsTeXFormat
 4, 285, 289, 291, 295, 302,
 306, 312, 318, 325, 332, 341–
 343, 350, 389, 390, 553, 588,
 594, 604, 605, 639, 806, 841,
 847, 857, 858, 865, 893, 1046, 1047
 \new@glossaryentry
 87, 88, 202, 433, 500,
 683, 751, 939, 1007, 1094, 1162
 \new@ifnextchar 75, 93, 94, 115, 116,
 141–145, 147–169, 258, 259,
 346–349, 427, 436, 437, 449,
 450, 468–480, 525, 526, 606–
 609, 677, 687, 688, 700, 701,
 719–732, 777, 778, 859–862,
 933, 943, 944, 957, 958, 973–
 986, 1032, 1033, 1087, 1098,
 1099, 1112, 1113, 1128–1142, 1188
 \newacronym 255, 255, 256, 260, 525,
 527, 538, 539, 541, 543, 545,
 547, 549, 550, 776–778, 790,
 792, 793, 795, 797, 800, 802,
 803, 1031, 1033, 1186, 1187, 1189
 \newacronymhook 260,
 261, 387, 527, 538, 540, 541,
 543, 545, 547, 549, 550, 582,
 778, 779, 790, 792, 794, 796,
 798, 800, 802, 804, 835, 1033, 1189
 \newacronymstyle 262, 263–268,
 270, 272, 273, 528–532, 534–
 536, 780–784, 786–788, 1035–
 1039, 1041–1043, 1190–1196, 1198
 \newcommand 4–
 6, 8–30, 32–34, 36–38, 40–43,
 46–49, 51–72, 74–82, 84–87,
 89–94, 96–98, 104, 105, 107–
 112, 114–118, 120, 122–128,
 130, 131, 138–184, 190, 193–
 198, 200, 203–209, 211–218,
 220, 227–233, 236, 237, 239,

240, 242–251, 254–260, 262–
 264, 273, 275, 277, 279–285,
 287, 288, 290–292, 305, 306,
 313, 332, 333, 338, 344–349,
 351, 356–360, 377, 386–389,
 391–401, 403–405, 407–440,
 444–451, 453, 455–459, 461,
 467–488, 492, 494–497, 499,
 501–516, 518–528, 530, 536–
 554, 557–559, 574, 582, 587,
 588, 594, 605–610, 612, 619,
 623, 634, 637, 641–650, 652,
 653, 655–691, 695–701, 704,
 706–710, 712, 718–739, 743,
 744, 746–751, 753–781, 788–
 807, 810–812, 827, 835, 840,
 841, 847, 859–866, 872, 876,
 888, 891, 893, 895–917, 919–
 947, 951–958, 960–966, 972–
 994, 998, 1001–1006, 1008–
 1029, 1031–1036, 1043–1064,
 1066–1072, 1074–1102, 1105–
 1110, 1112, 1113, 1115–1121,
 1127–1149, 1154, 1157–1161,
 1164–1179, 1181–1192, 1199–1202
 \newcount 17,
 85, 396, 432, 645, 646, 682,
 899, 900, 938, 1054, 1055, 1092
 \newcounter 14,
 15, 394, 395, 644, 898, 899, 1053
 \NewDocumentCommand 138, 1127
 \newenvironment
 245, 520, 771, 1027, 1182
 \newglossary 19, 20, 41,
 42, 76, 77, 199, 397, 398, 409,
 410, 427, 428, 498, 646, 647,
 659, 660, 677, 678, 750, 901,
 902, 914, 915, 933, 1005, 1055,
 1056, 1069, 1070, 1087, 1088, 1160
 \newglossaryentry 9, 42,
 84, 87, 87, 112, 261, 392, 410,
 431, 433, 448, 527, 537, 539,
 540, 542, 544, 546, 548, 550,
 582–586, 641, 660, 681, 683,
 698, 779, 789, 791, 793, 794,
 797, 799, 801, 803, 835–839,
 895, 915, 937, 939, 955, 1033,
 1050, 1070, 1092, 1094, 1110, 1189
 \newglossarystyle 251, 289, 292–
 305, 307–317, 319–331, 333–
 338, 340, 523, 611–637, 639,
 775, 864, 866–892, 1029, 1184
 \newif 6, 7, 22, 30, 33, 37,
 112, 210, 390, 398, 403, 406,
 407, 504, 639, 640, 648, 653,
 655, 657, 756, 893, 894, 902,
 907, 910, 912, 955, 1012, 1048,
 1057, 1062, 1064, 1066, 1110, 1167
 \newignoredglossary
 78, 428, 678, 934, 1088
 \newlabel 138, 1127
 \newlength 138, 295, 296, 307, 318,
 326, 336, 467, 613, 614, 620,
 626, 627, 630, 631, 636, 718,
 867, 873, 880, 884, 890, 972, 1127
 \newrobustcmd 8, 9,
 71, 72, 87, 88, 93–97, 115, 116,
 126, 141–171, 175, 181, 182,
 215, 216, 249, 252, 256–259,
 277, 337, 346–349, 360–363,
 391, 392, 424, 425, 433, 437,
 449, 450, 458, 468–486, 507,
 525, 526, 559–561, 605–609,
 637, 641, 674, 675, 683, 687,
 688, 700, 701, 709, 719–736,
 738, 759, 777, 778, 812–814,
 859–862, 890, 895, 930, 931,
 939, 943–946, 957, 958, 962,
 963, 973–987, 989, 992, 993,
 1014, 1015, 1028, 1029, 1031–
 1033, 1044, 1050, 1084–1086,
 1094, 1098–1101, 1112, 1113,
 1117, 1128–1143, 1145, 1148,
 1170, 1183, 1185, 1187, 1188, 1199
 \newterm 42, 410, 660, 915, 1070
 \newtoks 132, 194, 260,
 462, 495, 526, 713, 746, 778,
 967, 1001, 1033, 1121, 1157, 1189
 \newwrite 36, 88, 184, 189, 194, 198,
 407, 433, 488, 491, 495, 498,
 656, 684, 739, 742, 747, 749,
 911, 939, 994, 997, 1001, 1004,
 1066, 1094, 1149, 1152, 1157, 1160
 \nfss@text 8, 391, 641, 895, 1050
 \noalign 305, 619, 872
 \nobreak 294,
 306, 595, 613, 619, 848, 866, 873
 \noexpand 21, 47, 59, 60, 89, 103, 104,
 107, 129, 131, 132, 137, 138,
 180, 191–193, 199, 208, 213,

216, 220, 221, 229, 230, 253, 255, 261, 398, 411, 417, 418, 434, 443, 444, 457, 460, 461, 466, 467, 485, 492–498, 504, 505, 507, 509–512, 520, 521, 525, 527, 537, 539, 540, 542, 544, 546, 548, 550, 553, 554, 582–586, 588, 648, 661, 667, 668, 684, 694, 695, 707, 708, 710–713, 718, 737, 744–750, 755–757, 759, 761–764, 772, 773, 776, 777, 779, 789, 791, 793–795, 797, 799, 801, 803, 806, 807, 835–839, 842, 902, 917, 923, 924, 940, 950, 951, 953, 964–966, 971, 972, 992, 998–1005, 1011, 1013, 1015, 1017, 1019, 1030, 1031, 1033, 1034, 1057, 1071, 1077, 1078, 1095, 1105–1107, 1119–1121, 1126, 1127, 1147, 1154–1156, 1158, 1159, 1161, 1166, 1169, 1170, 1173–1175, 1186, 1187, 1189	nowarn (option) 23
noglossaryindex (option) 42	\ns@ACRfull 258, 259, 525, 526, 777, 1032, 1188
nogroupskip (option) 14	\ns@Acrfull 258, 525, 777, 1032, 1188
\nohyperpage 252, 523, 775, 1029, 1185	\ns@acrfull 258, 525, 777, 1032, 1188
nohypertypes (option) 23	\ns@ACRfullpl 259, 526, 778, 1033, 1188
\noindent 314–316, 318, 335–337, 521, 624–626, 636, 637, 773, 877–880, 889, 890	\ns@Acrfullpl 259, 526, 778, 1033, 1188
\noist 188, 190, 193 , 194, 495, 593, 594, 746, 846, 847, 997, 998, 1001, 1152, 1154, 1157	\ns@acrfullpl 259, 526, 777, 778, 1032, 1188
nolangwarn (option) 23	\ns@ACRlong 167, 479, 731, 985, 1140
nolist (option) 13	\ns@Acrlong 166, 479, 730, 984, 1140
nolong (option) 12	\ns@acrlong 165, 479, 730, 984, 1140
nomain (option) 20	\ns@ACRlongpl 169, 480, 732, 986, 1142
nonumberlist (key) 82	\ns@Acrlongpl 168, 480, 732, 986, 1141
nonumberlist (option) 12	\ns@acrlongpl 167, 480, 731, 985, 1141
\nopostdesc 42 , 51 , 51 , 98 , 219 , 410, 413, 440, 509, 594, 660, 663, 691, 761, 847, 915, 919, 947, 1016, 1070, 1074, 1101, 1172	\ns@ACRshort 163, 477, 729, 982, 1138
nopostdot (option) 13	\ns@Acrshort 162, 477, 728, 982, 1138
noredefwarn (option) 24	\ns@acrshort 161, 162, 476, 728, 982, 1137
\normalbaselineskip 305 , 619 , 872	\ns@ACRshortpl 165, 478, 730, 984, 1139
nostyles (option) 13	\ns@Acrshortpl 164, 478, 729, 983, 1139
nosuper (option) 13	\ns@acrshortpl 163, 477, 729, 983, 1138
notranslate (option) 30	\ns@newglossary 76 , 76 , 427, 677, 933, 1087
notree (option) 13	\null 131– 138, 191–193, 220, 461–467, 492–494, 509, 712–718, 744– 746, 761, 965–972, 998–1001, 1017, 1120–1127, 1154–1156, 1172
	\number 16, 89, 102, 107, 114, 208, 211, 396, 434, 442, 449, 503–505, 521, 554, 645, 684, 693, 700, 755, 756, 773, 807, 899, 940, 949, 953, 957, 1010, 1012, 1054, 1095, 1104, 1107, 1112, 1166–1168
	numberedsection (option) 10
	\numberline 58, 416, 666, 922, 1077
	numberline (option) 10
	numbers (option) 41
	\numexpr 114, 449, 700, 957, 1112
	O
	\O 27, 402, 651, 905, 1060
	\o 27, 402, 651, 905, 1060
	\OE 27, 401, 651, 905, 1060
	\oe 27, 401, 651, 905, 1060
	\oldacronym 254 , 524, 776, 1031, 1186
	\openout 36, 88, 185, 189, 194, 407, 433, 488, 491, 495, 590,

593, 656, 684, 739, 742, 747, 843, 846, 911, 940, 994, 997, 1002, 1066, 1094, 1149, 1152, 1157	
options:	
acronym	20
acronymlists	22
acronyms	20
automake	37
automakegloss	38
automakeglosslite	38
compatible-2.07	39
compatible-3.07	39
counter	22
counterwithin	14
debug	7
description	32
disablemakegloss	35
dua	33
entrycounter	14
esclocations	12
footnote	32
hyperfirst	32
index	41
indexonlyfirst	32
kernelglossredefs	40
languages	32
locales	32
makeindex	33
mfirstuc	42
noglossaryindex	42
nogroupskip	14
nohypertypes	23
nolangwarn	23
nolist	13
nolong	12
nomain	20
nonumberlist	12
noposteddot	13
noredefwarn	24
nostyles	13
nosuper	13
notranslate	30
notree	13
nowarn	23
numberedsection	10
numberline	10
numbers	41
order	33
restoremakegloss	35
sanitize	29
sanitizesort	28
savenumberlist	12
savewrites	39
section	10
seeautonumberlist	12
seenoindex	9
shortcuts	33
smallcaps	33
smaller	33
sort	15
style	11
subentrycounter	15
symbols	41
toc	10
translate	30
ucmark	14
writeglslabelnames	37
writeglslabels	36
xindy	34
xindygloss	34
xindynoglsnumbers	34
\OR	549, 802
\or	7, 9–11, 31, 37, 38, 40–42, 241, 242, 333, 391–393, 404, 409, 517, 598, 635, 640, 642, 654, 659, 769, 851, 888, 894, 896, 897, 908, 912, 914, 915, 1024, 1025, 1049, 1051, 1063, 1067, 1069, 1070, 1180
order (option)	33
\org@glossaryentrynumbers	219, 243, 509, 518, 761, 770, 1016, 1026, 1172, 1181
\org@glossarytitle	218, 219, 508, 509, 760, 1016, 1171, 1172
\org@glspostdescription	51, 413, 663, 919, 1074
\org@ifKV@glslink@hyper	127, 128, 459, 710, 963, 964, 1118, 1119
\outputpenalty	305, 306, 619, 872
P	
\p@	291, 312, 332, 612, 623, 634, 865, 876, 887, 888
\p@gls@hyp@opt	125, 125, 458, 709, 962, 1117
\PackageError	9, 18, 19, 28, 37, 47, 59, 66, 69, 70, 75, 81, 84, 92–98, 100, 101, 112, 123, 170, 193, 194,

199, 202–204, 227, 230, 231,
 233, 236, 241, 243, 250, 251,
 256, 262, 263, 363, 389, 392,
 397, 407, 411, 417, 422–424,
 427, 429, 431, 436–442, 448,
 457, 481, 494, 495, 498, 500,
 501, 512–514, 517, 518, 522,
 523, 528, 540, 541, 545, 548,
 561, 594, 641, 642, 646, 657,
 661, 667, 672–674, 677, 679,
 681, 686–690, 692, 693, 698,
 708, 733, 746, 747, 750, 751,
 753, 764–766, 769, 770, 774,
 775, 779, 780, 792, 793, 798,
 800, 801, 814, 847, 895, 896,
 900, 901, 906, 912, 917, 923,
 928–930, 932, 935, 937, 942–
 946, 948, 949, 955, 961, 987,
 1001, 1005, 1007, 1008, 1019–
 1021, 1024, 1025, 1029, 1031,
 1034, 1035, 1050, 1055, 1060,
 1067, 1072, 1078, 1082, 1084,
 1087, 1090, 1092, 1097–1103,
 1110, 1116, 1142, 1157, 1160,
 1162, 1164, 1175–1177, 1180,
 1181, 1184, 1185, 1187, 1190, 1191
 \PackageInfo 7,
 35, 194, 206, 390, 391, 406,
 407, 495, 503, 640, 641, 656,
 747, 754, 894, 895, 911, 1002,
 1009, 1049, 1065, 1066, 1157, 1165
 \PackageWarning
 7, 23, 43, 45, 390, 391,
 399, 640, 649, 894, 903, 1049, 1057
 \PackageWarningNoLine
 7, 23, 342, 390, 391,
 399, 588, 604, 640, 641, 649,
 841, 858, 894, 895, 903, 1049, 1058
 \pagegoal 306, 619, 873
 \pagelistname . . . 48, 51, 298–300,
 303, 310–312, 321–325, 329,
 331, 412, 413, 615–618, 621–
 623, 628–630, 632–634, 662,
 663, 869–871, 875, 876, 882–
 884, 886, 887, 917, 919, 1072, 1073
 \par 51, 291, 294, 312,
 314–318, 332, 333, 335–341,
 413, 521, 595, 599–601, 612,
 613, 623–626, 634–639, 663,
 773, 848, 852–854, 865, 866,
 876, 878–880, 887–893, 919, 1074
 parent (key) 82
 \parindent 313–
 318, 333–336, 338–341, 599–
 601, 623–626, 634–636, 638,
 639, 852–854, 877–880, 888–893
 \parskip
 313–316, 333, 334, 336, 623–
 625, 634–636, 877–879, 888–890
 \PassOptionsToPackage
 343, 350, 553, 605, 806, 858
 \pdfstringdefPreHook 209, 1011, 1167
 \penalty 305, 619, 872
 \PGLS 349, 608, 862
 \PglS 347, 347, 607, 860
 \pgls 346, 347, 606, 859
 \PGLSpl 349, 609, 862
 \PglSpl 348, 348, 608, 861
 \pglSpl 346, 348, 607, 860
 \phantomsection 56, 416, 666, 921, 1076
 plural (key) 80
 pluralaccess (key) 352
 \printacronyms 20, 398, 647, 901, 1056
 \PrintChanges . 6, 390, 640, 894, 1048
 \printglossaries 200, 218, 498, 508,
 750, 760, 1006, 1015, 1161, 1171
 \printglossary 20,
 24, 41, 42, 200, 217, 218, 243,
 398, 400, 409, 410, 498, 508,
 518, 647, 649, 659, 660, 750,
 759, 760, 770, 901, 903, 914,
 915, 1006, 1015, 1025, 1056,
 1058, 1069, 1070, 1161, 1171, 1181
 \printindex . . . 42, 410, 660, 915, 1070
 \printnoidxglossaries
 201, 218, 499, 508,
 751, 760, 1006, 1016, 1162, 1171
 \printnoidxglossary
 200, 201, 204, 218,
 218, 230, 231, 243, 499, 501,
 508, 512, 513, 518, 751, 753,
 760, 764, 765, 770, 1006, 1008,
 1015, 1016, 1020, 1021, 1025,
 1162, 1164, 1171, 1175, 1176, 1181
 \printnumbers . 41, 409, 659, 915, 1069
 \printsymbols . 41, 409, 659, 914, 1069
 \ProcessOptions
 343, 350, 553, 605, 806, 858
 \ProcessOptionsX
 43, 410, 660, 915, 1070

<code>\prop</code>	222–224	1180, 1182, 1184, 1186, 1189, 1199
<code>\protect</code> ..	58, 122, 171, 263–265, 271, 287, 368, 377–379, 383, 384, 416, 456, 529, 530, 534, 535, 538, 542, 544, 568, 569, 575, 576, 580, 666, 707, 780– 782, 786, 791, 794, 796, 821, 822, 828, 829, 833, 922, 961, 987, 1035–1037, 1041, 1046, 1077, 1116, 1143, 1191, 1192, 1197	<code>\protected@write</code>
<code>\protected@csedef</code>	95, 438, 689, 945, 1099	74, 76, 138, 185, 187, 198, 201, 204, 206, 216, 287, 426, 427, 488, 489, 498, 499, 502, 503, 507, 590, 610, 676, 677, 739– 741, 749–751, 753, 755, 759, 843, 863, 932, 933, 994, 995, 1004–1006, 1008, 1010, 1015, 1046, 1087, 1088, 1127, 1150, 1151, 1160, 1162, 1164, 1165, 1170
<code>\protected@csxappto</code>	1047	<code>\protected@xdef</code>
<code>\protected@csxdef</code>	94, 438, 688, 944, 1047, 1099	16–18, 21, 26, 85, 100, 101, 103, 104, 106, 212, 219, 220, 355, 356, 395, 396, 398, 401, 432, 445, 505, 556, 557, 645– 648, 650, 682, 692, 694, 696, 757, 761, 809, 810, 899, 900, 902, 905, 938, 948, 950, 952, 1012, 1016, 1054–1056, 1059, 1093, 1103, 1105–1107, 1168, 1172
<code>\protected@eappto</code>	36, 78, 105, 208, 657, 678, 695, 755, 911, 934, 951, 1011, 1066, 1089, 1106, 1166	<code>\providecommand</code>
<code>\protected@edef</code>	8, 11, 18, 21, 36, 60, 62, 65, 67, 71, 75, 76, 78, 79, 94–97, 99, 100, 104–106, 122, 123, 127, 129, 130, 171, 180–182, 197, 199, 201, 206, 210, 212, 213, 216, 227, 228, 242, 244, 250, 253, 261, 277, 285, 359, 360, 387, 391, 393, 418, 419, 421, 423, 441, 444, 445, 451–453, 456, 460, 485, 503–505, 517, 520– 523, 527, 545, 550, 553, 554, 559, 590, 606, 609, 641, 642, 646, 648, 657, 668, 669, 671, 673, 674, 677, 678, 688–692, 695, 696, 702, 703, 707, 710, 711, 737, 738, 751, 754, 756, 757, 759, 762, 763, 769, 771– 775, 779, 798, 803, 806, 807, 812, 835, 843, 844, 859, 863, 895, 897, 900, 902, 911, 924, 925, 927, 928, 930, 932–934, 944–948, 951, 952, 961, 963– 965, 987, 991, 992, 1004–1006, 1010, 1012, 1013, 1015, 1017, 1018, 1025, 1026, 1029, 1030, 1033, 1044, 1046, 1050, 1051, 1055, 1057, 1066, 1078, 1080, 1082–1084, 1087–1089, 1099– 1103, 1106, 1116, 1118–1120, 1143, 1147, 1148, 1160–1162, 1165, 1167–1170, 1173, 1174,	5, 20, 35, 47, 48, 56, 74, 114, 140, 187, 198, 201, 221, 291, 312, 332, 345, 351, 398, 406, 411, 412, 415, 426, 449, 468, 489, 498, 499, 502, 509, 510, 520, 521, 554, 606, 612, 623, 634, 647, 656, 661, 662, 665, 676, 700, 719, 741, 749, 751, 754, 761, 762, 772, 807, 859, 865, 876, 887, 893, 901, 910, 917, 921, 932, 957, 973, 995, 1004–1006, 1017, 1048, 1056, 1065, 1072, 1076, 1087, 1112, 1128, 1151, 1160, 1162, 1173
		<code>\provideglossaryentry</code>
		87, 433, 683, 939, 1094
		<code>\ProvidesFile</code> .
		49, 412, 662, 917, 1072
		<code>\ProvidesGlossariesLang</code>
		49, 412, 662, 917, 1072
		<code>\ProvidesPackage</code>
		4, 285, 289, 291, 295, 302, 306, 312, 318, 325, 332, 341–343, 350, 389, 390, 553, 588, 594, 604, 605, 609, 611, 612, 614, 617, 620, 623, 626, 630, 634, 639, 806, 841, 847, 857, 858, 862, 864, 865, 867, 871, 873, 876, 880, 884, 887, 893, 1046, 1047

Q	
<code>\quark</code>	223
R	
<code>\r</code>	27, 401, 651, 905, 1060
<code>\raggedright</code>	304, 305, 307–312, 326–331, 618– 623, 631–634, 872–876, 884–887
<code>\raisebox</code> . . .	138, 467, 719, 972, 1127
<code>\ref</code>	244, 519, 771, 1026, 1182
<code>\refstepcounter</code>	107, 108, 244, 519, 770, 771, 952, 953, 1026, 1107, 1108, 1182
<code>\regex</code>	249, 1028, 1184
<code>\relax</code>	7, 10, 13, 17, 18, 20, 30–32, 36, 37, 40, 42, 45, 46, 61, 74, 80, 82, 86, 89, 101, 103, 109, 113, 114, 125, 126, 129, 130, 132–137, 172, 187–191, 193, 194, 197, 199, 201, 203, 207, 212, 213, 215, 218, 220, 237, 238, 241, 249, 250, 278, 287, 291, 292, 305, 312, 317, 318, 332, 333, 335–341, 354, 355, 369, 370, 390, 393, 394, 396, 397, 404, 407, 409–411, 419, 426, 429, 430, 432, 434, 441– 443, 446, 448, 449, 458, 460– 466, 482, 490–492, 494, 495, 497–501, 503, 505, 507–509, 515, 517, 522, 523, 551, 556, 569, 570, 583–586, 589, 592– 594, 598–601, 610, 612, 619, 623, 626, 634–640, 642, 643, 645–647, 653, 654, 656, 657, 659–661, 669, 676, 679, 680, 682, 684, 692, 694, 697, 699, 700, 709, 711–718, 733, 741– 744, 746, 749–753, 755, 757– 761, 767, 769, 774, 804, 809, 822, 823, 836–839, 842, 843, 845–847, 851–854, 863, 865, 866, 872, 876, 879, 880, 887– 894, 896–898, 900, 901, 908, 909, 911, 912, 914–916, 924, 932, 935, 936, 938, 940, 948, 950, 954, 956, 957, 962, 964, 965, 967–972, 987, 988, 996– 1001, 1004–1008, 1010, 1012– 1014, 1016, 1017, 1022, 1024, 1028, 1029, 1044, 1047, 1049, 1051, 1052, 1054–1056, 1062– 1064, 1066–1071, 1079, 1087, 1089, 1090, 1093, 1095, 1103, 1105, 1108, 1111, 1112, 1117, 1119, 1120, 1122–1126, 1143, 1151, 1152, 1154, 1156, 1157, 1160–1163, 1166, 1168, 1170– 1172, 1178, 1180, 1184, 1200
<code>\renewacronymstyle</code> 263, 377–381, 383, 385, 386, 528, 575–577, 579–582, 780, 828–830, 832–835, 1035, 1190
<code>\renewcommand</code>	5–7, 9–15, 18– 20, 22–24, 29, 31, 32, 35, 37– 39, 41, 42, 46, 50–53, 66, 78, 81–83, 98, 108, 112–114, 180, 181, 183, 184, 190, 191, 197, 199–203, 219, 221, 225, 241, 242, 256, 260, 261, 263–273, 289, 290, 292–300, 302–304, 306–323, 326–328, 330, 333– 341, 351, 354, 355, 363, 366, 368, 376–380, 382–387, 390– 395, 397–400, 402–410, 412– 414, 422, 428–430, 440, 448, 449, 485–487, 492, 493, 497– 501, 509, 510, 517, 527–536, 538–541, 543, 545, 547, 549, 550, 554, 556, 562, 565, 567, 568, 574–586, 588, 589, 594– 600, 602–604, 611–629, 631– 649, 652, 654–660, 662–664, 672, 678–680, 691, 698–700, 737–739, 743, 744, 749–752, 761, 762, 769, 778–788, 790, 792, 793, 795–798, 800, 802, 803, 807, 809, 815, 818, 820, 821, 827–839, 841, 842, 847– 853, 855–857, 864–882, 884– 904, 906, 908, 909, 911– 916, 918–920, 928, 934–936, 947, 953, 955–957, 992–994, 998, 999, 1004–1008, 1016, 1017, 1024, 1025, 1031, 1033– 1043, 1048–1053, 1055–1058, 1061, 1063–1065, 1067–1070, 1072–1074, 1082, 1089–1091, 1101, 1108, 1110–1112, 1147– 1149, 1154, 1155, 1160–1163,

<code>\SetFootnoteAcronymStyle</code>	<code>short</code> (key)	83
. 277 , 545 , 549 , 797 , 802	<code>short-long</code> (acrstyle)	265 , 378
<code>\SetGenericNewAcronym</code>	<code>short-long-desc</code> (acrstyle)	267 , 380
. 260 , 262 , 527 , 528 ,	<code>shortaccess</code> (key)	352
778 , 780 , 1033 , 1034 , 1189 , 1190	<code>shortcuts</code> (option)	33
<code>\setglossarypreamble</code>	<code>shortplural</code> (key)	84
. 54 , 414 , 664 , 920 , 1075	<code>shortpluralaccess</code> (key)	353
<code>\setglossarysection</code>	<code>\show</code>	279–284 , 387–389 ,
. 56 , 415 , 665 , 921 , 1076 551–553 , 587 , 588 , 804–806 ,	
<code>\setglossarystyle</code> 840 , 841 , 1044–1046 , 1200–1202	
. 218 , 241 , 250 , 278 , 293–	<code>\showacronymlists</code>	
. 295 , 297–305 , 308–317 , 319– 283 , 552 , 805 , 1046 , 1201	
. 325 , 327–331 , 334 , 335 , 337 ,	<code>\showglocounter</code>	
. 340 , 508 , 522 , 523 , 551 , 613– 280 , 551 , 804 , 1045 , 1200	
. 637 , 639 , 760 , 774 , 804 , 866–	<code>\showglodesc</code> 281 , 552 , 805 , 1045 , 1201	
. 872 , 874–890 , 892 , 1016 , 1024 ,	<code>\showglodescaccess</code>	388 , 587 , 840
. 1029 , 1044 , 1171 , 1180 , 1184 , 1200	<code>\showglodescplural</code>	
<code>\setglossentrycompatibility</code> 281 , 552 , 805 , 1045 , 1201	
. 247 , 517 , 521 , 523 , 769 , 773 , 774	<code>\showglodescpluralaccess</code>	
<code>\setkeys</code> 388 , 587 , 840	
. 29 , 34 ,	<code>\showglofield</code> 283 , 552 , 805 , 1046 , 1201	
. 46 , 56 , 100 , 128 , 181 , 182 , 219 ,	<code>\showglofirst</code> 279 , 551 , 804 , 1044 , 1200	
. 261 , 403 , 406 , 411 , 415 , 441 ,	<code>\showglofirstaccess</code>	388 , 587 , 840
. 459 , 486 , 509 , 527 , 538 , 539 ,	<code>\showglofirstpl</code>	
. 541 , 543 , 545 , 547 , 549 , 550 , 279 , 551 , 804 , 1044 , 1200	
. 653 , 655 , 661 , 665 , 692 , 710 ,	<code>\showglofirstpluralaccess</code>	
. 738 , 760 , 778 , 790 , 792 , 793 , 388 , 587 , 840	
. 796 , 798 , 800 , 802 , 803 , 907 ,	<code>\showgloflag</code> 283 , 552 , 805 , 1045 , 1201	
. 910 , 916 , 921 , 948 , 963 , 992 ,	<code>\showgloindex</code> 282 , 552 , 805 , 1045 , 1201	
. 1016 , 1033 , 1062 , 1065 , 1071 ,	<code>\showglolevel</code> 279 , 551 , 804 , 1044 , 1200	
. 1076 , 1102 , 1118 , 1148 , 1172 , 1189	<code>\showgloloclist</code>	
<code>\setlength</code> 283 , 552 , 805 , 1046 , 1201	
. 295 , 296 ,	<code>\showglolong</code> 282 , 552 , 805 , 1045 , 1201	
. 307 , 313–316 , 318 , 326 , 333 ,	<code>\showglolongaccess</code>	388 , 587 , 841
. 334 , 336 , 339 , 340 , 600 , 601 ,	<code>\showglolongpluralaccess</code>	
. 613 , 614 , 620 , 623–627 , 630 , 389 , 588 , 841	
. 631 , 634–636 , 638 , 854 , 867 ,	<code>\showgloname</code> 281 , 552 , 805 , 1045 , 1201	
. 873 , 877–880 , 884 , 888–890 , 892	<code>\showglonameaccess</code>	387 , 587 , 840
<code>\SetSmallAcronymDisplayStyle</code>	<code>\showgloparent</code>	
. 277 , 545 , 547 , 798 , 800 279 , 551 , 804 , 1044 , 1200	
<code>\SetSmallAcronymStyle</code>	<code>\showgloplural</code>	
. 277 , 547 , 549 , 800 , 802 279 , 551 , 804 , 1044 , 1200	
<code>\setStyleFile</code>	<code>\showglopluralaccess</code>	388 , 587 , 840
. 51 , 413 , 414 , 663 , 664 , 919 , 1074	<code>\showgloshort</code> 282 , 552 , 805 , 1045 , 1201	
<code>\settodepth</code>	<code>\showgloshortaccess</code>	388 , 587 , 840
. 108 , 953 , 1108 388 , 587 , 840	
<code>\settoheight</code>	<code>\showglosort</code> 282 , 552 , 805 , 1045 , 1201	
. 108 , 467 , 719 , 953 , 1108	<code>\showglossaries</code>	
<code>\settowidth</code> 283 , 552 , 805 , 1046 , 1201	
. 109 , 264 , 530 , 600 ,		
. 601 , 637 , 638 , 781 , 853 , 854 ,		
. 891 , 892 , 953 , 1036 , 1108 , 1192		
<code>\setupglossaries</code>		
. 45 , 410 , 660 , 916 , 1070		
<code>\sfcode</code>		
. 13 , 394 , 644 , 898 , 1052		

<code>\showglossarycounter</code>	525, 529–536, 542, 545, 546,
..... 284 , 553 , 806 , 1046 , 1202	568, 575–583, 589, 591, 592,
<code>\showglossaryentries</code>	595–603, 610, 612–614, 620,
..... 284 , 553 , 806 , 1046 , 1202	627, 631, 634–638, 641, 642,
<code>\showglossaryin</code>	656–659, 661, 667, 670–673,
..... 283 , 552 , 806 , 1046 , 1201	680, 681, 698, 700, 701, 707–
<code>\showglossaryout</code>	709, 737, 740–742, 746–751,
..... 284 , 553 , 806 , 1046 , 1202	753, 759, 760, 762, 771, 774,
<code>\showglossarytitle</code>	777, 780–788, 794, 798, 799,
..... 284 , 553 , 806 , 1046 , 1202	821, 822, 828–836, 842, 844–
<code>\showglosymbol</code>	846, 848–856, 864–866, 868,
..... 282 , 552 , 805 , 1045 , 1201	873, 874, 880, 884, 888–892,
<code>\showglosymbolaccess</code> ..	895, 896, 910–912, 914, 917,
388 , 587 , 840	923, 926–928, 932, 936, 937,
<code>\showglosymbolplural</code>	955, 957, 958, 961, 991, 994–
..... 282 , 552 , 805 , 1045 , 1201	997, 1001–1006, 1008, 1011,
<code>\showglosymbolpluralaccess</code>	1015–1017, 1026, 1032, 1035–
..... 388 , 587 , 840	1044, 1047, 1050, 1051, 1065–
<code>\showglotext</code> 279 , 551 , 804 , 1044 , 1200	1069, 1072, 1078, 1081, 1083,
<code>\showglotextaccess</code>	1087, 1090, 1092, 1110, 1112,
387 , 587 , 840	1113, 1116, 1147, 1150–1152,
<code>\showglotype</code> 280 , 551 , 804 , 1044 , 1200	1157–1162, 1164, 1167, 1170,
<code>\showglouseri</code> 280 , 551 , 804 , 1045 , 1200	1171, 1173, 1182, 1187, 1191–1199
<code>\showglouserii</code>	
..... 280 , 551 , 804 , 1045 , 1200	<code>\spacefactor</code> .. 13, 394, 644, 898, 1052
<code>\showglouseriii</code>	<code>\SS</code>
..... 280 , 551 , 804 , 1045 , 1200	27, 402, 651, 905, 1060
<code>\showglouseriv</code>	<code>\ss</code>
..... 281 , 551 , 805 , 1045 , 1200	27, 402, 651, 905, 1060
<code>\showglouserv</code> 281 , 552 , 805 , 1045 , 1200	<code>\str</code>
<code>\showglouservi</code>	250, 1028, 1184
..... 281 , 552 , 805 , 1045 , 1201	<code>\string</code>
<code>sm-short-long</code> (acrstyle)	9, 18, 19,
266	24, 28, 35, 37, 40, 47, 58–60,
<code>sm-short-long-desc</code> (acrstyle) 268 , 380	62–67, 74–76, 82, 84, 89, 90,
<code>smallcaps</code> (option)	93, 94, 105, 106, 112, 114–117,
33	130, 131, 133, 134, 137, 138,
<code>smaller</code> (option)	179, 183–190, 193, 194, 198–
33	204, 208, 213, 214, 218, 221,
<code>\SmallNewAcronymDef</code>	230, 231, 243, 256, 277, 287,
277 ,	389, 392, 397, 400, 406–409,
387 , 546 , 547 , 586 , 799 , 800 , 839	411, 417–423, 426, 427, 430,
<code>sort</code> (key)	431, 434, 436, 437, 445, 448–
79	450, 457, 458, 460, 461, 463,
<code>sort</code> (option)	464, 466, 485, 487–492, 494,
15	495, 498–502, 506–510, 512,
<code>\space</code> 9, 35, 37, 40, 47, 59, 63, 64,	513, 518, 521, 523, 588–594,
66, 67, 75, 82, 84, 112, 115–	610, 641, 642, 646, 649, 656–
117, 122, 179, 185–189, 193,	659, 661, 667–673, 676, 677,
195–197, 199–201, 203, 204,	680, 681, 685, 687, 688, 695,
208, 216, 218, 221, 245, 257,	696, 698, 700, 701, 707–709,
263–273, 277, 288, 290, 291,	711, 712, 714, 715, 717, 718,
293, 294, 296, 307, 319, 326,	737–744, 746, 747, 749–753,
332–336, 338–340, 368, 377–	757, 758, 760–762, 764, 765,
386, 392, 406–409, 411, 417,	770, 773, 774, 802, 841–847,
420–423, 430, 431, 448, 450,	
456–458, 485, 488–491, 494–	
501, 507, 508, 510, 519, 523,	

863, 895, 896, 900, 901, 903,	
906, 910–912, 914, 917, 922–	
928, 932, 933, 936, 937, 941,	
943, 944, 951, 952, 955, 957,	
958, 965–969, 971, 991, 993–	
998, 1000, 1001, 1004–1009,	
1011, 1013, 1014, 1016, 1017,	
1020, 1021, 1025, 1031, 1044,	
1046, 1050, 1051, 1055, 1058,	
1060, 1065–1069, 1072, 1077,	
1078, 1080–1083, 1087, 1088,	
1090, 1092, 1095, 1096, 1098,	
1106, 1107, 1110, 1112, 1113,	
1120, 1122–1124, 1126, 1127,	
1147–1154, 1156, 1157, 1160–	
1162, 1164, 1167–1169, 1171,	
1173, 1175, 1176, 1181, 1187, 1200	
<code>\strut</code>	293–296,
298, 299, 307, 309, 311, 319,	
321, 323, 326, 328, 330, 336,	
521, 595–598, 600, 602–604,	
612–616, 620–622, 627–629,	
631–633, 637, 773, 848–851,	
853, 855–857, 866–869, 874,	
875, 880–882, 884, 885, 887, 890	
style (option)	11
styles:	
<code>altlist</code>	293
<code>altlistgroup</code>	294
<code>altlisthypergroup</code>	294
<code>altlong4col</code>	301
<code>altlong4col-booktabs</code>	304
<code>altlong4colborder</code>	301
<code>altlong4colheader</code>	301
<code>altlong4colheaderborder</code>	301
<code>altlongragged4col</code>	310
<code>altlongragged4col-booktabs</code>	305
<code>altlongragged4colborder</code>	311
<code>altlongragged4colheader</code>	311
<code>altlongragged4colheaderborder</code>	
.	312
<code>altsuper4col</code>	324
<code>altsuper4colborder</code>	324
<code>altsuper4colheader</code>	324
<code>altsuper4colheaderborder</code>	325
<code>altsuperragged4col</code>	329
<code>altsuperragged4colborder</code>	331
<code>altsuperragged4colheader</code>	330
<code>altsuperragged4colheaderborder</code>	
.	331
<code>alttree</code>	338
<code>alttreegroup</code>	340
<code>alttreehypergroup</code>	340
<code>index</code>	333
<code>indexgroup</code>	334
<code>indexhypergroup</code>	334
<code>inline</code>	289
<code>list</code>	292
<code>listdotted</code>	294
<code>listgroup</code>	293
<code>listhypergroup</code>	293
<code>long</code>	296
<code>long-booktabs</code>	302
<code>long3col</code>	297
<code>long3col-booktabs</code>	303
<code>long3colborder</code>	298
<code>long3colheader</code>	298
<code>long3colheaderborder</code>	299
<code>long4col</code>	299
<code>long4col-booktabs</code>	303
<code>long4colborder</code>	300
<code>long4colheader</code>	300
<code>long4colheaderborder</code>	300
<code>longborder</code>	297
<code>longheader</code>	297
<code>longheaderborder</code>	297
<code>longragged</code>	307
<code>longragged-booktabs</code>	304
<code>longragged3col</code>	309
<code>longragged3col-booktabs</code>	304
<code>longragged3colborder</code>	309
<code>longragged3colheader</code>	310
<code>longragged3colheaderborder</code>	310
<code>longraggedborder</code>	308
<code>longraggedheader</code>	308
<code>longraggedheaderborder</code>	308
<code>mcolalttree</code>	316
<code>mcolalttreegroup</code>	317
<code>mcolalttreehypergroup</code>	317
<code>mcolalttreespannav</code>	317
<code>mcolindex</code>	313
<code>mcolindexgroup</code>	313
<code>mcolindexhypergroup</code>	313
<code>mcolindexspannav</code>	313
<code>mcoltree</code>	314
<code>mcoltreegroup</code>	314
<code>mcoltreehypergroup</code>	314
<code>mcoltreenoname</code>	315
<code>mcoltreenonamegroup</code>	315
<code>mcoltreenonamehypergroup</code>	316

<code>\textmd</code>	254, 524, 776, 1031, 1186	<code>\theglossaryentry</code> 14, 245, 394, 519,	
<code>\textrm</code>	254, 524, 776, 1031, 1186	644, 771, 898, 1026, 1053, 1182	
<code>\textsc</code>	254, 265, 266, 272,	<code>\theglossarysubentry</code>	
	385, 524, 530, 531, 535, 540, 15, 245, 395, 519,	
	543, 545, 547, 581, 776, 782,	644, 771, 899, 1026, 1053, 1182	
	787, 792, 796, 798, 800, 834,	<code>\theglentrycounter</code>	
	1031, 1037, 1042, 1186, 1193, 1198	129,	
<code>\textsf</code>	254, 524, 776, 1031, 1186	209, 212, 460, 504, 505, 590,	
<code>\textsl</code>	254, 524, 776, 1031, 1186	710, 711, 756, 757, 843, 964,	
<code>\textsmaller</code> 266, 272, 385, 530, 531,		965, 1011, 1012, 1119, 1167, 1168	
	536, 540, 543, 545, 547, 581,	<code>\theH</code>	214, 506, 758, 1013, 1169
	782, 783, 787, 792, 796, 798,	<code>\theHglossaryentry</code>	
	800, 834, 1037, 1042, 1193, 1198 14, 394, 644, 898, 1053	
<code>\texttt</code>	254, 524, 776, 1031, 1186	<code>\theHglossarysubentry</code>	
<code>\textulc</code> . . .	256, 525, 777, 1031, 1187 15, 395, 644, 899, 1053	
<code>\textup</code>	254, 256, 524,	<code>\theHglentrycounter</code>	
	525, 776, 777, 1031, 1186, 1187 129, 209, 212, 460,	
<code>\TH</code>	27, 402, 651, 905, 1060	504, 505, 711, 756, 757, 964,	
<code>\th</code>	27, 402, 651, 905, 1060	965, 1011, 1012, 1119, 1167, 1168	
<code>\the</code>	47, 50, 60, 65, 67, 76,	<code>\thepage</code>	138, 1127
	132–138, 185, 189, 191, 193,	<code>\thesection</code> . . .	47, 411, 661, 916, 1071
	205, 206, 211, 216, 237, 253,	<code>\this@dialect</code>	50, 341,
	261–265, 269, 270, 377, 378,	342, 412, 413, 588, 604, 605,	
	382, 383, 387, 411, 412, 418,	662, 663, 841, 857, 858, 918, 1073	
	421, 423, 427, 462–467, 488,	<code>\tiny</code>	8, 391, 641, 895, 1050
	491–494, 502–505, 507, 515,	<code>\tl</code>	5, 44,
	520, 521, 524, 527–530, 533,	108, 139, 221, 223–226, 232–	
	534, 537, 539, 540, 542, 544,	235, 249, 893, 1048, 1108, 1127	
	546–548, 550, 553, 554, 575,	<code>toc (option)</code>	<u>10</u>
	578, 580, 582–586, 588, 590,	<code>\toks@</code>	47, 50, 60, 65, 67, 76,
	593, 661, 662, 668, 671, 673,	132–138, 191–193, 216, 253,	
	677, 713–718, 739, 742, 744,	411, 412, 418, 421, 423, 427,	
	746, 754, 756, 759, 767, 772,	462–467, 492–494, 507, 520,	
	773, 776, 779–781, 785, 786,	521, 524, 553, 554, 588, 661,	
	789–791, 793–795, 797, 799,	662, 668, 671, 673, 677, 713–	
	801, 803, 806, 807, 828, 831,	718, 744–746, 759, 772, 773,	
	833, 835–839, 841, 843, 846,	776, 806, 807, 841, 917, 918,	
	917, 918, 923, 927, 928, 933,	923, 927, 928, 933, 967–972,	
	967–972, 994, 997, 999–1001,	998–1001, 1015, 1030, 1071,	
	1009, 1010, 1012, 1015, 1022,	1073, 1078, 1082, 1083, 1088,	
	1030, 1033–1036, 1040, 1041,	1121–1127, 1154, 1156, 1170, 1186	
	1071, 1073, 1078, 1082, 1083,	<code>\toprule</code>	302, 303, 617, 618, 871
	1088, 1121–1127, 1150, 1152,	<code>\TrackIfKnownLanguage</code> .	49, 917, 1072
	1154, 1156, 1165, 1167, 1170,	<code>\trans@languages</code>	
	1178, 1186, 1189–1192, 1195, 1197 49, 412, 662, 918, 1072	
<code>\the@numberlist</code>		<code>\translate</code>	50, 51,
	180, 485, 486, 737, 991, 992, 1147	412, 413, 662, 663, 918, 919, 1073	
<code>\theglossary</code> . .	6, 390, 640, 894, 1048	<code>translate (option)</code>	<u>30</u>
<code>theglossary (env.)</code>	<u>245</u>	<code>\translatelet</code>	
		19, 20, 41, 397, 398, 409, 647,	
		659, 901, 902, 914, 915, 1056, 1069	

	Y		Z
<code>\year</code> 185, 189, 488, 491, 590, 593, 739, 742, 843, 846, 994, 997, 1150, 1152	<code>\Z</code> 249, 1028, 1184
		<code>\z@</code> 306, 619, 872, 873