

Documented Code For glossaries v5.1

Nicola L.C. Talbot

Dickimaw Books

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

2026-04-21

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	51
2.4	Xindy	61
2.5	Loops and conditionals	70
2.6	Defining new glossaries	79
2.7	Defining new entries	83
2.8	Resetting and unsetting entry flags	111
2.9	Keeping Track of How Many Times an Entry Has Been Unset . .	116
2.10	Loading files containing glossary entries	121
2.11	Using glossary entries in the text	122
	2.11.1 Links to glossary entries	127
	2.11.2 Displaying entry details without adding information to the glossary	174
2.12	Adding an entry to the glossary without generating text	185
2.13	Creating associated files	187
2.14	Writing information to associated files	209
2.15	Glossary Entry Cross-References	221
2.16	Displaying the glossary	223
2.17	Acronyms	268
2.18	Predefined acronym styles	274
2.19	Predefined Glossary Styles	292
2.20	Debugging Commands	293
2.21	Compatibility with version 2.07 and below	299

3	Glossary Styles	299
3.1	Glossary hyper-navigation definitions (glossary-hypernav.sty) . . .	299
3.2	In-line Style (glossary-inline.sty)	303
3.3	List Style (glossary-list.sty)	306
3.4	Glossary Styles using longtable (glossary-long.sty)	310
3.5	Glossary Styles using longtable and booktabs (glossary-longbooktabs.sty)	316
3.6	Glossary Styles using longtable (glossary-longragged.sty)	321
3.7	Glossary Styles using multicol (glossary-mcols.sty)	327
3.8	Glossary Styles using supertabular environment (glossary-super.sty)	334
3.9	Glossary Styles using supertabular environment (glossary-superragged.sty)	341
3.10	Tree Styles (glossary-tree.sty)	348
4	Multi-Lingual Support	419
4.1	Babel Captions (glossaries-babel.sty)	419
4.2	Polyglossia Captions (glossaries-polyglossia.sty)	420
5	Prefix Support (glossaries-prefix Code)	421
6	Accessibility Support (glossaries-accsupp Code)	428
6.1	Defining Replacement Text	430
6.2	Accessing Replacement Text	434
6.3	Displaying the Glossary	454
6.4	Acronyms	456
6.5	Debugging Commands	466
7	Backwards Compatibility	467
7.1	glossaries-compatible-207	467
7.2	glossaries-compatible-307	467
8	Rollback	468
8.1	Rollback v4.46 (glossaries-2020-03-19.sty)	468
8.2	Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)	631
8.3	Rollback v4.46 (glossaries-babel-2020-03-19.sty)	666
8.4	Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)	667
8.5	Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)	672
8.6	Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)	683
8.7	Rollback v4.46 (glossaries-prefix-2020-03-19.sty)	683
8.8	Rollback v4.46 (glossary-hypernav-2020-03-19.sty)	688
8.9	Rollback v4.46 (glossary-inline-2020-03-19.sty)	689
8.10	Rollback v4.46 (glossary-list-2020-03-19.sty)	690
8.11	Rollback v4.46 (glossary-list-2020-03-19.sty)	692
8.12	Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)	695
8.13	Rollback v4.46 (glossary-longragged-2020-03-19.sty)	698
8.14	Rollback v4.46 (glossary-mcols-2020-03-19.sty)	701
8.15	Rollback v4.46 (glossary-super-2020-03-19.sty)	705
8.16	Rollback v4.46 (glossary-superragged-2020-03-19.sty)	709

8.17	Rollback v4.46 (glossary-tree-2020-03-19.sty)	712
8.18	Rollback v4.49 (glossaries-2021-11-01.sty)	718
8.19	Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)	884
8.20	Rollback v4.49 (glossaries-babel-2021-11-01.sty)	919
8.21	Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)	920
8.22	Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)	926
8.23	Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)	936
8.24	Rollback v4.49 (glossaries-prefix-2021-11-01.sty)	936
8.25	Rollback v4.49 (glossary-hypernav-2021-11-01.sty)	941
8.26	Rollback v4.49 (glossary-inline-2021-11-01.sty)	942
8.27	Rollback v4.49 (glossary-list-2021-11-01.sty)	944
8.28	Rollback v4.49 (glossary-long-2021-11-01.sty)	946
8.29	Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)	949
8.30	Rollback v4.49 (glossary-longragged-2021-11-01.sty)	952
8.31	Rollback v4.49 (glossary-mcols-2021-11-01.sty)	955
8.32	Rollback v4.49 (glossary-super-2021-11-01.sty)	958
8.33	Rollback v4.49 (glossary-superragged-2021-11-01.sty)	962
8.34	Rollback v4.49 (glossary-tree-2021-11-01.sty)	966
8.35	Rollback v4.52 (glossaries-2022-11-03.sty)	971
8.36	Rollback v4.52 (glossary-hypernav-2022-11-03.sty)	1125
9	Rollback	1126
9.1	Rollback v4.54 (glossaries-2024-04-03.sty)	1126
9.2	Rollback v4.58 (glossaries-2025-03-19.sty)	1280
9.3	Rollback v4.58 (glossary-mcols-2025-03-19.sty)	1446
9.4	Rollback v4.58 (glossary-tree-2025-03-19.sty)	1449

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` v5.1: L^AT_EX2e Package to Assist Generating Glossaries”.

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

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

INSTALL Installation instructions.

CHANGES Change log.

README.md Package summary.

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

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

2 Main Package Code (glossaries.sty)

2.1 Package Definition

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareRelease{v4.52}{2022-11-03}{glossaries-2022-11-03.sty}
\DeclareRelease{v4.54}{2024-04-03}{glossaries-2024-04-03.sty}
\DeclareRelease{v4.58}{2025-03-19}{glossaries-2025-03-19.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossaries}[2026/04/21 v5.1 (NLCT)]
```

Required packages:

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

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

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

```
\ExplSyntaxOn
```

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

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

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

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

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

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

Automatically identify exclusions, blockers and mappings.

```
\glsmfuexcl
```

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

```
\glsmfublocker
```

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

```
\glsmfuaddmap
```

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

```
}
```

```
{
```

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

```
\RequirePackage{textcase}
```

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

```
\MFUsentencecase
```

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

Provide support for exclusions with `\MFUsentencecase`.

```
\glsmfuexcl
```

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

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

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

Just treat blockers and mappings as exclusions.

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

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

```
}
\ExplSyntaxOff
\RequirePackage{xfor}
```

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}%
}
```

`seenindex`

```
\define@choicekey{glossaries.sty}{seenindex}%
[\gls@seenindex@val\gls@seenindex@nr]{error,warn,ignore}{%
  \ifcase\gls@seenindex@nr
  \renewcommand*{\@gls@see@noindex}{%
    \PackageError{glossaries}%
    {'\gls@xr@key' key may only be used after \string\makeglossaries\space
    or \string\makenoidxglossaries}%
  }
```

```

        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a ‘\gls@xr@key’ key}%
    }%
\or
    \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{‘\gls@xr@key’ key ignored}%
    }%
\or
    \renewcommand*{\@gls@see@noindex}{}%
\fi
}

```

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

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

```

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

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

```

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

```

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

```

section The `section` key can be used to set the sectional unit. If no unit is specified, use `section` as the default. The starred form of the named sectional unit will be used. New to v4.9: `none` indicates no sectional unit.

```

\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph,none}[section]{%
    \renewcommand*{\@@glossarysec}{#1}}

```

Determine whether or not to use numbered sections.

\@@glossarysecstar

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

```

\@@glossaryseclabel

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

```

\@glossary@write@seclabel

```

\newcommand*{\@glossary@write@seclabel}[1]{%
    \@provide@glossary@seclabel
    \protected@write\@auxout{}{\string\@glossarylabeltype{#1}{\currentglossary}}%
}

```

```

\@gls@currentglossaryxrlabel
\newcommand*{\@gls@currentglossaryxrlabel}{}

\@provide@glossary@seclabel
\newcommand*{\@provide@glossary@seclabel}{%
\protected@write\@auxout{}\string\providecommand\string\@glossarylabeltype[2]{}%
\let\@provide@glossary@seclabel\relax
}

\@glossarylabeltype Information for external processors. The first argument is the label automati-
cally added via \@glossaryseclabel and the second argument is the glossary
type. This won't pick up glossaries where the label has been insert via some
other means (for example, by adding \label to the glossary preamble).
\newcommand*{\@glossarylabeltype}[2]{}

\glsautoprefix Prefix to add before label if automatically generated:
\newcommand*{\glsautoprefix}{}

numberedsection
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\renewcommand*{\@gls@currentglossaryxrlabel}{%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\renewcommand*{\@gls@currentglossaryxrlabel}{%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\label{\glsautoprefix\@glo@type}%
\@glossary@write@seclabel{\glsautoprefix\@glo@type}%
\renewcommand*{\@gls@currentglossaryxrlabel{\glsautoprefix\@glo@type}%
}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}%
\@glossary@write@seclabel{\glsautoprefix\@glo@type}%
\renewcommand*{\@gls@currentglossaryxrlabel{\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}%
}
```

Ideally, I would like to convert all the `xkeyval` keys to `l3keys` but this would have to be co-ordinated with the same update in `glossaries-extra` and could cause problems. So for now, only the new style options use `l3keys`.

style-options Note that the options won't be available until the applicable package has been loaded.

```
\ExplSyntaxOn
\define@key{glossaries.sty}{style-options}{
\keys_set:nn { glossaries / styles } { #1 }
}
\keys_define:nn { glossaries / styles }
{
unknown .code:n =
{
No ~ key=value ~ style ~ support ~ for ~ ' \l_keys_key_str '. ~
(The ~ applicable ~ glossary ~ style ~ package ~
must ~ be ~ loaded ~ before ~ adjusting ~ the ~ style.)
}
}
\ExplSyntaxOff
```

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}{%
  \ifglsentrycounter
    Define the glossaryentry counter if it doesn't already exist.
    \ifundef\c@glossaryentry
    {%
      \ifx\@gls@counterwithin\@empty
        \newcounter{glossaryentry}%
      \else
        \newcounter{glossaryentry}[\@gls@counterwithin]%
      \fi
      \def\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]{}
\glssentrycounterfalse
```

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}%
  \glssentrycountertrue
  \@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.

`\ine@glossarysubentrycounter`

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

Check if counter already defined.

```
\ifundef\c@glossarysubentry
{%
  \ifglssubentrycounter
  \ifglssentrycounter
  \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 \@gls@makeindexapp\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 `<sort cs>` is the temporary control sequence containing the sort value before it has been sanitized and had `makeindex/xindy` special characters escaped.

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

`\@glo@check@sortallowed`

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

```
\ExplSyntaxOn
```

`\@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}%
}
```

Version v.48: switched to `\tl_gset:cV`:

```
\tl_gclear_new:c { glo###2@sort }
\tl_gset:cV { glo###2@sort } \@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@setsort@standard
```

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

```
\newcommand*\glssortnumberfmt[1]{%
\int_compare:nNnT { #1 } < { 100000 } { 0 }
\int_compare:nNnT { #1 } < { 10000 } { 0 }
\int_compare:nNnT { #1 } < { 1000 } { 0 }
\int_compare:nNnT { #1 } < { 100 } { 0 }
\int_compare:nNnT { #1 } < { 10 } { 0 }
\int_eval:n { #1 }
}
```

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

```
\newcommand*\@gls@setsort@def{%
```

Store entry information when it's defined.

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

Defined count register associated with the glossary. Version 4.58: switched to L3

int variable (renamed `\glossary@<label>@sortcount` to `g__glossary_<label>_sortcount_int`).

```
\def\@gls@defsortcount##1{%
\int_new:c { g__glossary_ ##1 _sortcount_int }
}%
```

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.

```
\int_if_exist:cF { g__glossary_ ##1 _sortcount_int }
{
\@gls@defsortcount { ##1 }
}
\int_gincr:c { g__glossary_ ##1 _sortcount_int }
\tl_gclear_new:c { glo@ ##2 @sort }
\tl_gset:ce { glo@ ##2 @sort }
{
\exp_args:Nc \glssortnumberfmt
{ g__glossary_ ##1 _sortcount_int }
}
}%
```

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 integer variable associated with the glossary.

```
\def\@gls@defsortcount##1{%
\int_new:c { g__glossary_ ##1 _sortcount_int }
}%
```

Initialise the sort key to empty.

```
\def\@gls@defsort##1##2{%
\tl_gclear_new:c { glo@ ##2 @sort }
\tl_gset:cn { glo@ ##2 @sort } { }
}%
```

If the sort key hasn't been set, increment the counter associated with the glossary and set the sort key.

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

Get the parent, if one exists

```
\tl_if_empty:cF { glo@ ##1 @parent }
{
```

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

```
\exp_args:nv \@gls@setsort { glo@ ##1 @parent }
}
```

Set index information for this entry

```
\tl_if_empty:cT { glo@ ##1 @sort }
{
\int_gincr:c
{
g__glossary_
\tl_use:c { glo@ ##1 @type }
_sortcount_int
}
\tl_gset:ce { glo@ ##1 @sort }
{
\exp_args:Nc \glssortnumberfmt
{
g__glossary_
\tl_use:c { glo@ ##1 @type }
_sortcount_int
}
```

```

    }
  }
  \@glo@storeentry{##1}%
}
}%

```

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

```

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

```

```

\ExplSyntaxOff

```

`\@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[gls2]{main}{gls2}{gls2}{\glossaryname}%
  \else
    \newglossary{main}{gls}{gls}{\glossaryname}%
  \fi
}
```

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

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

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

`\glsdefaulttype`

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

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

`\acronymtype`

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

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

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

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

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

```

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

```

```

\printacronyms Define \printacronyms at the start of the document if acronym is set and
\printacronyms hasn't already been defined.
\AtBeginDocument{%
\ifglsacronym
\providecommand*{\printacronyms}[1][{}]{%
\printglossary[type=\acronymtype,#1]}%
\fi
}

```

```

\@gls@do@acronymsdef Set default value
\newcommand*{\@gls@do@acronymsdef}{%

```

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

```

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

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

```

```

\glsacronymlists Comma-separated list of glossary labels indicating which glossaries contain
acronyms.
\newcommand*{\@glsacronymlists}{%

```

`\@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.

`\glsetexpandfield`

```
\glsetexpandfield{<field>}
```

Sets field to always expand.

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

`\glsetnoexpandfield`

```
\glsetnoexpandfield{<field>}
```

Sets field to never expand.

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

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

```
\glsetexpandfield{type}
```

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

```
\glsetnoexpandfield{desc}
```

`\gls@assign@descplural@field`

```
\glsetnoexpandfield{descplural}
```

`\@gls@sanitizename`

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

`\gls@assign@name@field` Don't expand name by default.

```
\glsetnoexpandfield{name}
```

```

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

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

s@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 NB datatool-base
hasn't been loaded yet.
\ExplSyntaxOn
\bool_new:N \l__glossaries_noidx_defer_sort_preprocess_bool
\bool_set_false:N \l__glossaries_noidx_defer_sort_preprocess_bool
\newcommand*{\@gls@noidx@nosanitizesort}
{
  \bool_if:NF \l__glossaries_noidx_defer_sort_preprocess_bool

```

```

{
  \cs_if_exist:NTF \datatool_sort_preprocess:Nn
  {
    \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
      }
    }
  }
  {
    \@@gls@old@noidx@nosanitizesort
  }
}
\ExplSyntaxOff

```

`\gls@old@noidx@nosanitizesort` Old method.

```

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

```

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

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

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

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

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

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

\@gls@removedoption

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

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

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

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

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

preprocess-sort

```
\ExplSyntaxOn
\define@choicekey{glossaries.sty}{preprocess-sort}{true,false}[true]{%
  \tl_clear:N \@gls@noidx@setsanitizesort
```

```

\glssanitizesortfalse
\glsssetnoexpandfield { sortvalue }
\tl_if_eq:nnTF { #1 } { true }
{
  \bool_set_false:N \l__glossaries_noidx_defer_sort_preprocess_bool
}
{
  \bool_set_true:N \l__glossaries_noidx_defer_sort_preprocess_bool
}
}
\ExplSyntaxOff

```

sanitizesort

```

\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
}

```

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}%
      \glssetexpandfield{name}%
      \glssetexpandfield{symbol}%
      \glssetexpandfield{symbolplural}%
      \glssetexpandfield{desc}%
      \glssetexpandfield{descplural}%
    }%
  }%
  \setkeys[gls]{sanitize}{#1}%
}%
}
```

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

```
\newif\ifglstranslate
```

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

`\@gls@usetranslator`

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

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

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



```

    {#3}%
}

```

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

```

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

```

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

```

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

```

Set the default value:

```

\glstranslatefalse

```

```

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

```

`\@gls@requested@languages`

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

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

```

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

```

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}}
```

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

```
\newif\ifglxindy
```

The default is `makeindex`:

```
\glxindyfalse
```

`\@gls@makeindexapp`

```
\newcommand\@gls@makeindexapp{makeindex}
```

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

```
\@gls@declareoption{makeindex}{%  
  \renewcommand\@gls@makeindexapp{makeindex}%  
  \glxindyfalse  
}
```

`upmendex` Define package option to specify that `upmendex` will be used to sort the glossaries. NB there's currently no support for automatically setting the dictionary. This will need to be done by adding the required switch to the `upmendex` call, for example with `\GlsAddToMakeIndexOpts`.

```
\@gls@declareoption{upmendex}{%  
  \renewcommand\@gls@makeindexapp{upmendex}%  
  \glxindyfalse  
}
```

`mendex` As above but for `mendex`

```
\@gls@declareoption{mendex}{%  
  \renewcommand\@gls@makeindexapp{mendex}%  
  \glxindyfalse  
}
```

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

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

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

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

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

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

Define key to set the language

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

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

```
\ifdefined\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\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
    {\protected@eappto\@glsl@fulllist{\@ignored@glossaries}}%
  \forallgl@entries[\@glsl@fulllist]{\@gl@entry}%
  {%
    \ifdefempty{\@gl@entry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@gl@thisentryname{%
      \else
        \letcs{\@gl@thisentryname}{glo\@gl@detoklabel{\@gl@entry}\@name}%
        \@onelevel@sanitize\@gl@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@gl@entry\do@write@glslabelsname{\@gl@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}

```

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

```

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

```

```

\@write@glslabelsname

```

```

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

```

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

```

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

```

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

```

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

```

Provide a way of saving glossary group information.

```

\glswriteglossarygroup Syntax: {\group-label}{\group-title}
\newcommand\glswriteglossarygroup[2]{%

```

```

\glswriteglossarysubgroup Syntax: {\level}{\parent-entry}{\group-label}{\group-title}
\newcommand\glswriteglossarysubgroup[4]{%

```

saveglossarygroups Save glossary group information in the aux file (for example, for external processors to gather information or to pick up the information in the next run).

```
\define@choicekey{glossaries.sty}{saveglossarygroups}%
  [\gls@saveglossarygroups@val\gls@saveglossarygroups@nr]%
  {true,false}[true]%
  {%
    \ifcase\gls@saveglossarygroups@nr\relax
      \renewcommand\glswriteglossarygroup{%
        \@gls@writeglossarygroup
      }%
      \renewcommand\glswriteglossarysubgroup{%
        \@gls@writeglossarysubgroup
      }%
    \or
      \renewcommand\glswriteglossarygroup[2]{}%
      \renewcommand\glswriteglossarysubgroup[4]{}%
    \fi
  }
```

\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}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][ ]{\printglossary[type=symbols,##1]}%

```

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

```

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

```

numbers Create a “symbols” glossary type

```

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

```

Default is not to define the numbers glossary:

```

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

```

```

\@gls@numbersdef
    \newcommand*{\@gls@numbersdef}{%
        \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
        \newcommand*{\printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
    }
    Define hook to set the toc title when translator is in use.
    \newcommand*{\gls@tr@set@numbers@toctitle}{%
        \translatelet{\glossarytoctitle}{Numbers (glossaries)}%
    }%
}

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

noglossaryindex Counteract index if it happens to be globally used in the document class.
\@gls@declareoption{noglossaryindex}{%
    \let\@gls@do@indexdef\relax
}

Default is not to define index glossary:
\newcommand*{\@gls@do@indexdef}{\}

\@gls@indexdef \indexname isn't set by glossaries.
\newcommand*{\@gls@indexdef}{%
    \newglossary[ilg]{index}{ind}{idx}{\indexname}%
    \newcommand*{\printindex}[1][\]{\printglossary[type=index,##1]}%
    \newcommand*{\newterm}[2][\]{%
        \newglossaryentry{##2}%
        {type={index},name={##2},description={\nopostdesc},##1}}
    \let\@gls@do@indexdef\relax
}%

mfirstuc Similar to mfirstuc's expanded and unexpanded package options. Not providing
the other option.
\define@choicekey{glossaries.sty}{mfirstuc}{%
    [\gls@mfirstuc@val\gls@mfirstuc@nr]%
    {expanded,unexpanded}%
}%
\ifcase\gls@mfirstuc@nr
% expanded
\renewcommand{\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
\or
% unexpanded
\renewcommand{\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
\fi
}

```

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

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

Now process options passed to the package:

```
\ProcessOptionsX
```

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

If LuaLaTeX and automake set, need to load `shellesc`. Note that if automake is passed to `\setupglossaries` instead, then `shellesc` will need to be loaded explicitly. 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
```

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

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
```

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

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}
```

```
\ifglautomake
```

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 { \sys_shell_shipout:e { #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
  \ifgl Sentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglssubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@dosubentrycounterdef
}

```

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

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

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

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

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

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

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

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

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

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

```
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
```

2.3 Predefined Text

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

Main glossary title:

```
\glossaryname
\providecommand*\glossaryname{Glossary}
```

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

```
\acronymname
\providecommand*\acronymname{Acronyms}
```

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

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

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

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

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

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

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

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

```
\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*{\glsettoctitle}[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]{}%
```

```
\@glsmakeindexappname
  \newcommand*{\@glsmakeindexappname}[1]{}%
```

This command is the value of the `page_compositor` `makeindex` key. Again, any redefinition of this command must take place *before* `\writeist` otherwise it will have no effect. As from 1.17, use `\glsSetCompositor` instead of directly redefining `\glscompositor`.

```
\glscompositor
  \newcommand*{\glscompositor}{.}
```

`\glsSetCompositor` Sets the compositor.

```
\newcommand*{\glsSetCompositor}[1]{%
  \renewcommand*{\glscompositor}{#1}}
```

Only use before `\makeglossaries`

`\@onlypremakeg\glsSetCompositor`

(The page compositor is usually defined as a dash when using `makeindex`, but most of the standard counters used by L^AT_EX use a full stop as the compositor, which is why I have used it as the default.) If `xindy` is used `\glscompositor` only affects the `arabic-page-numbers` location class.

`\@glsAlphaCompositor` This is only used by `xindy`. It specifies the compositor to use when location numbers are in the form $\langle letter \rangle \langle compositor \rangle \langle number \rangle$. For example, if `\@glsAlphaCompositor` is set to “.” then it allows locations such as A.1 whereas if `\@glsAlphaCompositor` is set to “-” then it allows locations such as A-1.

`\newcommand*\@glsAlphaCompositor{\glscompositor}`

`\glsSetAlphaCompositor` Sets the alpha compositor.

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

Can only be used before `\makeglossaries`

`\@onlypremakeg\glsSetAlphaCompositor`

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

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

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

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

Only has an effect when used before `\makeglossaries`

`\@onlypremakeg\glsSetSuffixF`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

`\setglossarypreamble`

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

Code provided by Michael Pock.

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

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

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

`\glossarypostamble`

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

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

```
\ExplSyntaxOn
\NewDocumentCommand \glossarysection { o m }
{
  \IfValueTF { #1 }
  {
    \glossaries_section:nn { #1 } { #2 }
  }
  {
    \glossaries_section:nn { #2 } { #2 }
  }
}
```

New to v4.9:

```
\cs_new:Nn \glossaries_section:nn
{
  \tl_if_eq:NnF \@glossarysec { none }
  {
    \ifcsundef{phantomsection}
    {
      \@glossarysection { #1 } { #2 }
    }
    {
      \p@glossarysection { #1 } { #2 }
    }
  }
}
```

```

\glsglossarymark { \glossarytoctitle }
}
}
\ExplSyntaxOff

```

\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{\glsupercase{#1}}{\glsupercase{#1}}%
\else
\@mkboth{#1}{#1}%
\fi
}

```

The required sectional unit is given by **\@@glossarysec** which was defined by the section package option. The starred form of the command is chosen. As from v4.8, if you don't want a section header use **none**.

\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.

```
\@pglossarysection
\newcommand*{\@pglossarysection}[2]{%
  \gls@clearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%
    \@gls@toc{#1}{\@glossarysec}%
    \csname\@glossarysec\endcsname*{#2}%
  }%
}
Do automatic labelling if required
\@glossaryseclabel
}
```

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

```
\newcommand*{\gls@doclearpage}{%
  \ifthenelse{\equal{\@glossarysec}{chapter}}{%
    {%
      \ifcsundef{cleardoublepage}%
      {%
        \clearpage
      }%
    }%
  }%
}
```

```

    {%
      \ifcsdef{if@openright}%
      {%
        \if@openright
          \cleardoublepage
        \else
          \clearpage
        \fi
      }%
    }%
    \cleardoublepage
  }%
}%
{}%
}

```

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

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

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

```

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

```

2.4 Xindy

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

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

```

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

```

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

```

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

\@xdyattributes Define list of attributes (\string is used in case the double quote character has
been made active)
\ifglxsindy
  \edef\@xdyattributes{\string"default\string"%
\fi

\@xdyattributelist Comma-separated list of attributes.
\ifglxsindy
  \edef\@xdyattributelist{}%
\fi

\@xdylocref Define list of markup location references.
\ifglxsindy
  \def\@xdylocref{}
\fi

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

\GlsAddXdyCounters Need to know all the counters that will be used in location numbers for Xindy.
Argument may be a single counter name or a comma-separated list of counter
names.
\ifglxsindy
  \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]{%
        \glsnxindywarning\GlsAddXdyAttribute
    }
\fi

```

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

```

\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}%
}

```

`\GlsAddXdyAttribute` Adds an attribute.

```

\ifglxindy

```

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

```

    \newcommand*\@glssaddxdyattribute[2]{%

```

Add to xindy attribute list

```

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

```

Add to xindy markup location.

```

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

```

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

```

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

```

High-level command:

```

    \newcommand*\GlsAddXdyAttribute[1]{%

```

Add to comma-separated attribute list

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

First add the attribute without the counter name:

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

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

```
\for\@this@counter:=\@xdycounters\do{%
\protected@edef\gls@do@addxdyattribute{%
\noexpand\@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]{%
\glsnoxywarning\GlsAddXdyAttribute}
\fi
```

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

```
\ifglsxindy
\newcommand*{\@gls@addpredefinedattributes}{%
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperarm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}}
```



```

\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}

\GlsAddXdyAttribute{glignore}
}
\else
\let\@glx@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]{%
\glxnoindywarning\GlsAddXdyAlphabet}
\fi

```

This code is only required for xindy:

```
\ifglxindy
```

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

```

\newcommand*{\@glx@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 `\@glx@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\@glx@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@glx@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

\ty@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"%
}%

\ty@Lclass@arabic-page-numbers Arabic numbers (1, 2, ...).
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%

\ty@Lclass@alpha-page-numbers Lower case alphabetical (a, b, ...).
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%

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

\ty@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"%
}

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

```

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

```
\def\@xdyuserlocationdefs{}
```

`\@xdyuserlocationnames` List of additional user location names

```
\def\@xdyuserlocationnames{}
```

End of xindy-only block:

```
\fi
```

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

```
\ifglxindy
```

```
\newcommand\@xdycrossrefhook{}
```

```
\fi
```

`\GlsAddXdyLocation` `\GlsAddXdyLocation[<prefix-loc>]{<name>}{<definition>}` Define a new location called *<name>*. The definition must use xindy syntax. (Note that this doesn't check to see if the location is already defined. That is left to xindy to complain about.)

```
\ifglxindy
```

```
\newcommand*\GlsAddXdyLocation[3][{}]{%
```

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

```
\ifx\@gls@tmp\@empty
```

```
\edef\@xdyuserlocationdefs{%
```

```
\@xdyuserlocationdefs ^^J%
```

```
(define-location-class \string"#2\string"^^J\space\space
```

```
\space(:sep \string"{}\glsopenbrace\string" #3
```

```
:sep \string"\glsclosebrace\string"))
```

```
}%
```

```
\else
```

```
\edef\@xdyuserlocationdefs{%
```

```
\@xdyuserlocationdefs ^^J%
```

```
(define-location-class \string"#2\string"^^J\space\space
```

```
\space(:sep "\glsopenbrace"
```

```
#1
```

```
:sep "\glsclosebrace\glsopenbrace" #3
```

```
:sep "\glsclosebrace"))
```

```
}%
```

```
\fi
```

```
\edef\@xdyuserlocationnames{%
```

```
\@xdyuserlocationnames^^J\space\space\space
```

```
\string"#2\string"}%
```

```
}
```

Only has an effect before `\writeist`:

```
\@onlypremakeg\GlsAddXdyLocation
```

```
\else
```

```
\newcommand*\GlsAddXdyLocation[2]{%
```

```
\glsnoxindywarning\GlsAddXdyLocation}
```

```
\fi
```

```

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

Change the location order.

\GlsSetXdyLocationClassOrder
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnoxywarning\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]{%
\glsnoxywarning\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]{%
    \glsnnoxindywarning\GlsAddXdyStyle}
\fi

```

\GlsSetXdyStyles Reset the list of required styles

```

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

```

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

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

```

\def\@xdylanguage#1#2{}

```

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

```

\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glstypedefaulttype]{%
    \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][]{%
    \glsnnoxindywarning\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.

```
\ifglxsindy
  \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.

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

`\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}%
}
```

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

```
\forallglsentries[<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*{\forallglsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glist@#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@}%
  {%
    \forallglsentries[\@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@#1@out}{#3}{#2}%
}
```

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

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

Low-level command to test if glossary is empty. Unlike `\glstrifemptyglossary` provided by `glossaries-extra` this doesn't warn if the glossary doesn't exist.

```
\ExplSyntaxOn
\prg_new_conditional:Npnn \glossaries_if_glossary_empty:n #1
{ T, F, TF }
{
  \bool_lazy_or:nnTF
    { \tl_if_exist_p:c { glolist@ #1 } }
    { \tl_if_empty_p:c { glolist@ #1 } }
  {
    \prg_return_true:
  }
  {
    \tl_if_eq:cnTF { glolist@ #1 } { , }
    {
      \prg_return_true:
    }
    {
      \prg_return_false:
    }
  }
}
\ExplSyntaxOff
```

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 `\glstrytext` 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\forglsentries\expandafter
    [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs\glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
      {%
        \def\do@glshaschildren{#2}%
        \@endfortrue
      }%
    }%
  }%
  \do@glshaschildren
}%
}

```

```
\ifglshasparent{<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}%
}

```

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

```

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

```

Version 4.59: Test if entry has description and is not suppressed:

```

\ExplSyntaxOn
\prg_new_conditional:Npnn \glossaries_if_has_nonsuppressed_desc:n #1
{ p , T, F, TF }
{
  \bool_lazy_or:nnTF
  {
    \bool_not_p:n { \tl_if_exist_p:c { glo@\glsdetoklabel{#1}@desc } }
  }
  {
    \bool_lazy_or_p:nn
    {
      \tl_if_empty_p:c { glo@\glsdetoklabel{#1}@desc }
    }
    { \tl_if_eq_p:cN { glo@\glsdetoklabel{#1}@desc } \@no@post@desc }
  }
  { \prg_return_false: }
  { \prg_return_true: }
}
\ExplSyntaxOff

\ifglshassymbol \ifglshassymbol{<label>}{<true part>}{<false part>}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}

\ifglshaslong \ifglshaslong{<label>}{<true part>}{<false part>}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

\ifglshasshort \ifglshasshort{<label>}{<true part>}{<false part>}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@\glsdetoklabel{#1}@short}%
  \ifdefempty\@glo@short
  {#3}%

```

```

    {%
      \ifdefequal\@glo@short\@gls@default@value
        {#3}%
      {#2}%
    }%
  }

```

```

\ifglsfieldvoid{<field>}{<label>}{<true part>}{<false
part>}}

```

\ifglsfieldvoid

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

```

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

```

```

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

```

\ifglshasfield

```

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

```

First check supplied field label is defined.

```

    \ifdef\@glo@thisvalue
      {%

```

Is defined, so now check if empty.

```

        \ifdefempty\@glo@thisvalue
          {%

```

Is empty, so doesn't have field set.

```

            #4%
          }%
        }%
      }%

```

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

```

        \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
      }%
    }%
  }%
}

```

```

    }%
  }%
}%
{%

```

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

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

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

```

\ifdef\@gls@thisfield
{%

```

Is defined, so now check if empty.

```

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

```

Is empty so field hasn't been set.

```

    #4%
  }%
{%

```

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

```

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

```

Value is set to the default value.

```

    #4%
  }%
{%

```

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

```

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

```

Not defined.

```

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

```

```
\glscurrentfieldvalue
```

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

2.6 Defining new glossaries

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

`\@glo@types`

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

`\@gls@provide@newglossary` If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\newcommand*{\@gls@provide@newglossary{%
\protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%}
```

Only need to do this once.

```
\let\@gls@provide@newglossary\relax
}
```

`\@gls@defaulttype`

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

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

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

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

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

`\gls@doentryfmt`

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

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

```
\newcommand*{\@gls@forbidtexext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
```

```

{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
{%
}%
}

```

`\gls@gobbleopt` Discard optional argument.

```

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

```

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

```

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

```

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

`\newglossary`

```

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

```

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

```

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

```

`\ns@newglossary` Define the unstarred version.

```

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

```

Check if default has been set

```

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

```


Add this to the list of glossary types:

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

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

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

Store the file extensions:

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

Store the title:

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

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

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

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

Define sort counter if required:

```
\@gls@defsortcount{#2}%
```

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

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

`\altnewglossary`

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

Only define new glossaries in the preamble:

```
\@onlypreamble{\newglossary}
```

Only define new glossaries before `\makeglossaries`

`\@onlypremakeg\newglossary`

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

`\@newglossary`

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

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

`\@gls@setcounter`

`\def\@gls@setcounter#1[#2]{%`
`\expandafter\def\csname @glotype@#1@counter\endcsname{#2}%`

Add counter to xindy list, if not already added:

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

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

`\@gls@getcounter`

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

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

`\glsdefmain`

Define the “acronym” glossaries if required.

`\@gls@do@acronymsdef`

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

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

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

`\newcommand*{\newignoredglossary}[1]{%`
`\ifdefempty\@ignored@glossaries`
`{%`
`\protected@edef\@ignored@glossaries{#1}%`
`}%`

```

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

```

\@ignored@glossaries List of ignored glossaries.

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

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

```

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

```

2.7 Defining new entries

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

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

```

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

```

description The `description` key is usually only used in the glossary, but can be made to appear in the text by redefining `\glsentryfmt` or using `\defglsentryfmt`. The

description key is required when defining a new glossary entry. If a long description is required, use `\longnewglossaryentry` instead of `\newglossaryentry`.

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

descriptionplural

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

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

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

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

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

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

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

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

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

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

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

\@gls@default@value

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

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

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

symbolplural

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

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

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

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

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

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

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

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

`\gls@set@xr@key`

Assign a cross-reference key. This has a check for an empty value because with `glossaries-extra` and unrestricted document definitions, the `glsdefs` 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
}

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

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

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

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

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

```

```

\glsnoidxprenumberlist
    \newcommand{\glsnoidxprenumberlist}[1]{%
        \csuse{glo@glsdetoklabel{#1}@prenumberlist}}

\@gls@savenonumberlist The nonumberlist option isn't saved by default (as it just sets the prefix) which
                        isn't a problem when the entries are defined in the preamble, but causes a
                        problem when entries are defined in the document. In this case, the value needs
                        to be saved so that it can be written to the .glsdefs file.
    \newcommand*{\@gls@savenonumberlist}[1]{%

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

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

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

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

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

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

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

```

```

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

```

```

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

```

```

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

```

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

```

\newcommand*\@glsnodels{%
\PackageError{glossaries}
{%
description key required in \string\newglossaryentry\space
for entry '\@glo@label'%
}
}

```



```

}%
{%
    You haven't specified the entry description%
}%
}%

```

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

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

```

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

```

`\@glsdefaultsort` Define command to set default sort.

```

\newcommand*{\@glsdefaultsort}{\@glo@name}

```

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

```

\newcount\gls@level

```

`\@gls@noexpand@field`

```

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

```

`\@gls@noexpand@fields`

```

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

```

```

}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%

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

\@gls@expand@field

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

@gls@expand@fields

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

```

`\@gls@startswithexpandonce`

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

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

`\gls@assign@field`

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

```

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

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

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

```
\newcommand*{\gls@gldefs@addlabel}[1]{\listxadd{\gls@gldefs@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 `.gldefs` 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.gldefs
  }%
  {}%
}

```

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

```

\ifglentryexists{#1}%
{%
  \xifinlist{#1}{\gls@docdef@labellist}%
  {%

```

This entry has already been defined in the document environment.

```

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\newcommand*{\glsdefs@newdocentry}[2]{%
\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]{\@Glsentryfield{##1}{#1}}%
```

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

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

```

\@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
\expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%

```

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

```

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

```

Finally the all caps version:

```

\ifcsdef{@GLS@user@#1@}%
{%
\PackageError{glossaries}%
{Can't define '\string#7' as helper command
'\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @GLS@user@#1\endcsname}[2][\{
\new@ifnextchar[%
{\csuse{@GLS@user@#1@}{##1}{##2}}%
{\csuse{@GLS@user@#1@}{##1}{##2}[]}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
\@gls@field@link{##1}{##2}{\glsuppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
\expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%

```

Add mappings.

```

\glsmfuaddmap{#3}{#4}%
\glsmfuaddmap{#5}{#6}%
\glsmfublocker{#7}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}

```

\glsfieldxdef

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

```

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

```

\glsfielddedef

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

```

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

```

\glsfieldgdef

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

```

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

```

\glsfielddef

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

```

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

```

\glsfieldfetch

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

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

```

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

```

```
}%
}
```

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

\glsunexpandedfieldvalue

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

\ifglsfieldeq

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

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

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

\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}%
\egroup
}
}

```

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

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

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

```

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

```

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

`\gls@defglossaryentry`

Defines a new entry without checking if it already exists.

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

Prevent any further use of `\GlsSetQuote`:

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

Store label

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Set the default sort:

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

Set the default counter:

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

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

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

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

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

```
\@gls@initnonnumberlist
```

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

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

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

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

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

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

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

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

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

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

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

```
\@newglossaryentryprehook
```

Extract key-val information from third parameter:

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

Check there is a default glossary.

```
\ifundef\glsdefaulttype
```

```
{%
```

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

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

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

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

```
}%
```

```
{ }%
```

Assign type. This must be fully expandable

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

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

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

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

```
{%
```

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

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



```

        {You need to define a new glossary type, before making entries
        in it}%
    }%
    {%
Check if it's an ignored glossary
    \ifignoredglossary\@glo@type
    {%
The description may be omitted for an entry in an ignored glossary.
    \ifx\@glo@desc\@glsnodesc
    \let\@glo@desc\empty
    \fi
    }%
    {%
    }%
    \protected@edef\@glolist@{\csname glolist@\@glo@type\endcsname}%

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

Initialise level to 0.
    \gls@level=0\relax
Has this entry been assigned a parent?
    \ifx\@glo@parent\empty
Doesn't have a parent. Set \glo@<label>@parent to empty.
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
    \else
Has a parent. Check to ensure this entry isn't its own parent.
    \ifdefequal\@glo@label\@glo@parent%
    {%
    \PackageError{glossaries}{Entry '@@glo@label' can't be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
    }%
    {%
Check the parent exists:
    \ifglentryexists{\@glo@parent}%
    {%
Parent exists. Set \glo@<label>@parent.
    \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
    \@glo@parent}%
Determine level.
    \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
    \advance\gls@level by 1\relax

```

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

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

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

```
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
{%
```

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

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

Set the level for this entry

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

Define commands associated with this entry:

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

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

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

```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
  {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi

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

```

Set default numberlist if not defined:

```

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

```

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

```

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

```

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

```

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

```

```
\gls@assign@desc{\@glo@label}%
```

Set the sort key for this entry:

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

```
\def\@glo@symbol{\@glo@text}%
```

```
\ifx\@glo@symbol\@glo@symbol
```

```
\let\@glo@symbol\@glo@text
```

```
\fi
```

```
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
```

```
\expandafter
```

```
\gls@assign@field\expandafter
```

```
{\csname glo@\@glo@label @symbol\endcsname}
```

```
{\@glo@label}{symbolplural}{\@glo@symbolplural}%
```

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

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

```
\noexpand\global
```

```
\noexpand\let\expandafter\noexpand
```

```
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
```

```
}%
```

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

```
\noexpand\global
```

```
\noexpand\let\expandafter\noexpand
```

```
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
```

```
}%
```

```
\csname glo@\@glo@label @flagfalse\endcsname
```

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

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

```
{%
```

```
\csdef{glo@\@glo@label @index}{}%
```

```
}
```

```
{%
```

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

```
}%
```

Sort out any cross-referencing if required. Version 4.59: this needs to occur after the index internal field has been set.

```
\@glo@autosee
```

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@ssee}%
```

```

        {%
        \protected@edef\@do@glsssee{%
        \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
        \noexpand\expandafter\noexpand\@glsssee\noexpand\@glo@list{\@glo@label}}%
        \@do@glsssee
        }%
        \@glo@autoseehook
    }%

\@glo@autoseehook

\newcommand*{\@glo@autoseehook}{-}

\@newglossaryentryprehook Allow extra information to be added to glossary entries:
\newcommand*{\@newglossaryentryprehook}{-}

\@newglossaryentryposthook Allow extra information to be added to glossary entries:
\newcommand*{\@newglossaryentryposthook}{-}

newglossaryentry@defcounters

\newcommand*{\@newglossaryentry@defcounters}{-}

\glsmoveentry Moves entry whose label is given by first argument to the glossary named in the
second argument.
\newcommand*{\glsmoveentry}[2]{%
\protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
\protected@edef\glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
\def\glo@list{,}%
\for\glssentries[\glo@type]{\glo@label}%
{%

\ifdefequal\@glo@thislabel\glo@label
    {}{\protected@eappto\glo@list{\glo@label,}}%
}%
\cslet\glolist@\glo@type{\glo@list}%
\csdef{glo@\@glo@thislabel @type}{#2}%
}

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

\@glossarysubentryfield Indicate what command should be used to display each subentry in the glossary.
(This enables the glossaries-accsupp package to use \accsuppglossarysubentryfield
instead.)

```

```

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

```

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

\@glo@storeentry

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

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

Escape makeindex/xindy special characters in the label:

```

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

```

Get the sort string and escape any special characters

```

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

```

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

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

Get the parent, if one exists

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

Write the information to the glossary file.

```
\ifglxindy
```

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

`\glsmeasureheight`

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

`\glsmeasuredepth`

```
\newcommand{\glsmeasuredepth}[2]{%
  \settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
```

`\glsmeasurewidth`

```
\newcommand{\glsmeasurewidth}[2]{%
  \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 `\cgl`s. (Only global unset is used if previous count was one as it doesn't make sense to have a local unset here given that the previous count was global.)

```
\def\@cgl#1##2[##3]{%
\ifnum\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\@cglspl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cglsplformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@cglspl@{##1}{##2}[##3]%
  \fi
}%
```

First letter uppercase singular:

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

First letter uppercase plural:

```
\def\@cGlspl@##1##2[##3]{%
  \ifnum\glentryprevcount{##2}=1\relax
    \cGlsplformat{##2}{##3}%
    \glunset{##2}%
  \else
    \@Glspl@{##1}{##2}[##3]%
  \fi
}%
```

Write information to aux file at the end of the document

```
\AtEndDocument{\@gls@write@entrycounts}%
```

Fetch previous count information from aux file. (No check here to determine if the entry is still defined.)

```
\renewcommand*{\@gls@entry@count}[2]{%
  \csgdef{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\glentrycurrcount{##1}+1}%
}
```

```
s@local@increment@currcount
```

```

\newcommand*{\@gls@local@increment@currcount}[1]{%
  \csedef{glo@glsetoklabel{#1}@currcount}{%
    \number\numexpr\glsetrycurrcount{#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{\@glsetry}{%
    \ifglsetry{\@glsetry}{%
      {\immediate\write\@auxout
        {\string\@gls@entry@count{\@glsetry}{\glsetrycurrcount{\@glsetry}}}}%
      }%
    }%
  }

```

\@gls@entry@count Default behaviour is to ignore arguments. Activated by `\glsetryenableentrycount`.

```

\newcommand*{\@gls@entry@count}[2]{}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

\newcommand*{\cglsformat}[2]{%
  \ifglshaslong{#1}{\glsetrylong{#1}}{\glsetryfirst{#1}}#2%
}

```

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

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

Add sentence case mapping.

```
\glsmfuaddmap{\cgl}{\cGls}
```

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

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

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

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

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

```
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
```

`\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]%
}
```


`\cglspformat` 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*{\cglspformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#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}
```

Add sentence case mapping.

```
\glsmfuaddmap{\cglsp1}{\cGlsp1}
```

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

```
\newcommand*{\@cGlsp1}[2][{}]{%
  \new@ifnextchar[{\@cGlsp1@{#1}{#2}}{\@cGlsp1@{#1}{#2}[]}%
}
```

`\@cGlsp1@` Read in the final optional argument. This defaults to same behaviour as `\Glsp1` but issues a warning.

```
\def\@cGlsp1@#1#2[#3]{%
  \GlossariesWarning{\string\cGlsp1\space is defaulting to
    \string\Glsp1\space since you haven't enabled entry counting}%
  \@Glsp1@{#1}{#2}[#3]%
}
```

`\cGlspformat` 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*{\cGlspformat}[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`, `\glsp1` and

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

uppercase variants or `\glsadd` and `\glsaddall` will appear in the glossary). The mandatory argument is the filename (with or without `.tex` extension).

```
\loadglsentries
    \newcommand*{\loadglsentries}[2][\@gls@default]{%
        \let\@gls@default\glsdefaulttype
        \def\glsdefaulttype{#1}\input{#2}%
        \let\glsdefaulttype\@gls@default
    }
```

`\loadglsentries` can only be used in the preamble:

```
\@onlypreamble{\loadglsentries}
```

2.11 Using glossary entries in the text

Any term that has been defined using `\newglossaryentry` (or `\newacronym`) can be displayed in the text (i.e. outside of the glossary) using one of the commands defined in this section. Unless you use `\glslink`, the way the term appears in the text is determined by `\glsdisplayfirst` (if it is the first time the term has been used) or `\glsdisplay` (for subsequent use). Any formatting commands (such as `\textbf`) is governed by `\glstextformat`. By default this just displays the *link text* “as is”.

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

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

```
\newcommand*{\glsentryfmt}{\glsgenentryfmt}
```

`\glsgenentryfmt` Define a generic format that just uses the first, text, plural or first plural keys (or the custom text) with the insert text appended.

```
\newcommand*{\glsgenentryfmt}{%
    \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 `\acrfullformat`, `\firstacronymfont` and `\acronymfont`.

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

Subsequent use:

```
\glsifplural  
{%
```

Subsequent plural form:

```
\glscapscase  
{%
```

Subsequent plural form, don't adjust case:

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

Subsequent plural form, make first letter upper case:

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

Subsequent plural form, all caps:

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

Subsequent singular form

```
\glscapscase  
{%
```

Subsequent singular form, don't adjust case:

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

Subsequent singular form, make first letter upper case:

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

Subsequent singular form, all caps:

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

First use:

```
\glsifplural  
{%
```

First use plural form:

```
\glscapscase  
{%
```

First use plural form, don't adjust case:

```
\genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, make first letter upper case:

```
\Genplacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use plural form, all caps:

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

First use singular form

```
\glscapscase
{%
```

First use singular form, don't adjust case:

```
\genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, make first letter upper case:

```
\Genacrformat{\glslabel}{\glsinsert}%
}%
{%
```

First use singular form, all caps:

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

User supplied text.

```
\glscustomtext
}%
}
```

`\genacrformat`

```
\genacrformat{\label}{\insert}
```

The full format used by `\glsgenacfmt` (singular).

```
\newcommand*{\genacrformat}[2]{%
\glsentrylong{#1}#2\space
```

```
(\protect\firstacronymfont{\glsentryshort{#1}})%
}
```

`\Genacrfullformat`

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

As above but makes the first letter upper case.

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

`\genplacrfullformat`

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

The full format used by `\glsngenacfmt` (plural).

```
\newcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
```

`\Genplacrfullformat`

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

As above but makes the first letter upper case.

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

```
\newcommand*{\p@glshyp@opt}[1][]{%  
  \let\glslinkvar\@thirdofthree  
  \@glshyp@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}{%  
  \@glshyp@opt\@glsc@link  
}
```

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

```
\newcommand*{\@glsc@link}[3][]{%  
  \glsdoifexistsordo{#2}%  
  {%  
    \let\do@glsc@link@checkfirsthyper\relax  
    \@glsc@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}{%  
  \@glshyp@opt\@Gls@link  
}
```

```

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

\glspostlinkhook
\newcommand*{\glspostlinkhook}{}

\@gls@link@checkfirsthyper Check for first use and switch off hyper key if hyperlink not wanted. (Should be
off if first use and hyper=false is on or if first use and both the entry is in an
acronym list and the acrfootnote setting is on.) This assumes the glossary type
is stored in \glsstype and the label is stored in \glslabel.
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glsstype
    \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@gl:disablehyperinlist Disable hyperlink if in the “nohyper” list.
\newcommand*{\do@gl:disablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glsstype}{\@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@gl:disablehyperinlist
Macros must set this before calling \@gls@link. The commands that check
the first use flag should set this to \@gls@link@checkfirsthyper otherwise it
should be set to \relax.
        \do@gls@link@checkfirsthyper
        \setkeys{glslink}{#1}%
Add a hook for the user to customise things after the keys have been set.
        \glslinkpostsetkeys
Store the entry’s counter in \theglsentrycounter
        \@gls@saveentrycounter
Define sort key if necessary:
        \@gls@setsort{\glslabel}%
(De-tok’ing done by \@do@wrglossary)
        \glsencapwrcontent{\@do@wrglossary{#2}}%
        \ifKV@glslink@hyper
            \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
        \else
            \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
        \fi

```

Restore original setting

```
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
\@gl@restore@gl@local
}
```

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

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

`\glolinkprefix`

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

`\glsentrycounter` Set default value of entry counter

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

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

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

Are we using equation counter?

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

If we're in align environment, `\xatlevel@` will be defined. (Can't test for `\@currentenv` as may be inside an inner environment.)

```
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gl@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gl@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gl@counter\endcsname}%
}
```

Check if hyperref version of this counter

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

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

Not using equation counter so no special measures:

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

Check if hyperref version of this counter

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

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

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

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

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

```

\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@sufffix{glsnumberformat}%
\else
\def\@glo@sufffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@sufffix{#1#2}%
\fi
\fi}

```

`\@gls@escbsdq` Escape backslashes and double quote marks. The argument must be a control sequence.

```

\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%

```

Unsanitize `\gls@numberpage`, `\gls@alphpage`, `\gls@Alphpage` and `\gls@romanpage` (thanks to David Carlisle for the suggestion.) Skip if `esclocations=false`.

```

\ifglsclocations
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dosubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dosubst
}%
\fi

```

Assign to required control sequence

```

\let#1=\gls@xdystring
}

```

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

```

\@gls@checkmkidxchars
\newcommand{\@gls@checkmkidxchars}[1]{%

```

```

\ifglxindy
  \@gls@escbsdq{#1}%
\else
  \def\@gls@checkedmkidx{%
    \expandafter\@gls@checkquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkescquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescactual#1\@nil"??\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkactual#1\@nil??\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkbar#1\@nil||\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkescbar#1\@nil|||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checklevel#1\@nil!!\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checkesclevel#1\@nil!!\!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                  \def\@gls@checkedmkidx{%
                    \edef\@do@gls@checkesc{%
                      \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
                      \@backslashchar\@backslashchar\noexpand\@nil
                    }%
                    \do@gls@checkesc
                    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                  }%
                }%
              }%
            }%
          }%
        }%
      }%
    }%
  }%
\fi
}

```

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\@nil{%
    \@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
\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
}
```

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

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

```

\fi
\@gls@xdycheckquote
}

```

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

```

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

```

Now go ahead and define \@gls@xdycheckbackslash

```

\def@gls@xdycheckbackslash

```

\glsdohypertarget

```

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

```

```

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

```

`\glsdohypertargethook` Hook for `\glsdohypertarget`

```
\newcommand*{\glsdohypertargethook}[2]{}

```

`\glslabelhypertarget` Provided for use in the definition of `\glsdohypertargethook` if required.

```

\NewDocumentCommand\glslabelhypertarget{mm}{%
\protected@write\@auxout
{\glslabelhypertargetdefs}%
{%
\string\newlabel
{\glslabelhypertargetprefix#1}%
{%
\glslabelhypertargetvalue}{\thepage}{#2}{#1}{}}%
}%
}

```

```
\ExplSyntaxOn

```

`\glslabelhypertargetvalue`

```

\newcommand\glslabelhypertargetvalue{
\bool_lazy_and:nnT
{ \tl_if_exist_p:N \glscurrententrylabel }
{ \bool_not_p:n { \tl_if_empty_p:N \glscurrententrylabel } }
{
\glsentryname \glscurrententrylabel
}
}
\ExplSyntaxOff

```

`\glslabelhypertargetprefix`

```
\newcommand*\glslabelhypertargetprefix{}

```

`\glslabelhypertargetdefs`

```

\newcommand*\glslabelhypertargetdefs{%
\let\glossentryname\glsentryname
\let\glossentrydesc\glsentrydesc
\let\glossentrysymbol\glsentrysymbol
\let\Glossentryname\glsentryname
\let\Glossentrydesc\glsentrydesc
\let\Glossentrysymbol\glsentrysymbol
\let\glossentrynameother\@gls@entry@field
}

```

```

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

```

`\glsdohyperlinkhook` Hook for `\glsdohyperlink`

```

\newcommand*{\glsdohyperlinkhook}[2]{}

```

`\glsdonohyperlink`

```

\newcommand*{\glsdonohyperlink}[2]{#2}

```

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

```

\ifcsundef{hyperlink}%
{%
    \let\@glslink\glsdonohyperlink
}%
{%
    \let\@glslink\glsdohyperlink
}

```

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

```

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

```

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

```

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

```

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

`\glsenablehyper`

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

Provide some convenience commands if not already defined:

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

Syntax:

```
\gls[options]{label}[insert text]
```

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

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

First determine which version is being used:

`\gls`

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

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

`\@gls`

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

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

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

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }
```

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

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

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

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

Indicate that this entry has now been used

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

```
\glspostlinkhook
}
```

`\Gls` behaves like `\gls`, but the first letter of the link text is converted to uppercase (note that if the first letter has an accent, the accented letter will need to be grouped when you define the entry). It is mainly intended for terms that start a sentence:

`\Gls`

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

Add sentence case mapping.

```
\glsmfuaddmap{\gls}{\Gls}
```

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

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

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

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

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
  }
```

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

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

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

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

Indicate that this entry has now been used

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

\glspostlinkhook
}
```

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

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

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

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

\@Gls@ Read in the final optional argument:

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

    \let\glsifplural\@secondoftwo
    \let\glsupcase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
```

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

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

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

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

Indicate that this entry has now been used

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

\glspostlinkhook
}
```

\glspl behaves in the same way as \gls except it uses the plural form.

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

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

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

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

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

    \let\glsifplural\@firstoftwo
    \let\gls caps case\@firstofthree
    \let\gls custom text\@empty
    \def\glsinsert{#3}%

```

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

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

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

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

Indicate that this entry has now been used

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

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

`\Glspl` behaves in the same way as `\glspl`, except that the first letter of the link text is converted to uppercase (as with `\Gls`, if the first letter has an accent, it will need to be grouped).

`\Glspl`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glspl}{\Glspl}
```

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

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

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

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

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

Determine what the link text should be (this is stored in `\@glo@text`). This needs to be expanded so that the `\@glo@text` can be passed to `\xmakefirstuc`. Note that `\@gls@link` sets `\glstype`.

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

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

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

Indicate that this entry has now been used

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

\glspostlinkhook
}
```

`\GLSpl` behaves like `\glspl` except that all the link text is converted to uppercase.

`\GLSpl`

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

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

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

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

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

    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
  }%
  \def\glsinsert{#3}%
```

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

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

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

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

Indicate that this entry has now been used

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

```
}%
```

```
\glspostlinkhook
```

```
}
```

`\glsdisp` `\glsdisp[<options>]{<label>}{<text>}` This is like `\gls` except that the link text is provided. This differs from `\glslink` in that it uses `\glsdisplay` or `\glsdisplayfirst` and unsets the first use flag.

First determine if we are using the starred form:

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

Defined the un-starred form.

```
\@glsdisp
```

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

```
\glsdoifexists{#2}{%
```

```
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
```

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

```
\let\glscapscase\@firstofthree
```

```
\def\glscustomtext{#3}%
```

```
\def\glsinsert{}%
```

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

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

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

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

Indicate that this entry has now been used

```
\@gls@do@glset{#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{<options>}{<label>}{<link text>}
```

`\@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}{\glstentrytext{#2}{#3}}%
}
```

`\GLStext` behaves like `\glstext` except the text is converted to uppercase.

`\GLStext`

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

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

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

Read in the final optional argument:

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

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

`\Glstext`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glstext}{\Glstext}
```

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

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

Read in the final optional argument:

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

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

`\glsfirst`

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

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

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

Read in the final optional argument:

```
\def\@glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
```

`\Glsfirst` behaves like `\glsfirst` except it displays the first letter in uppercase.

`\Glsfirst`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirst}{\Glsfirst}
```


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

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

Read in the final optional argument:

```
\def\@Glsfirst@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
```

`\GLSfirst` behaves like `\Glsfirst` except it displays the text in uppercase.

`\GLSfirst`

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

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

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

Read in the final optional argument:

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

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

`\glsplural`

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

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

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

Read in the final optional argument:

```
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
```

`\Glsplural` behaves like `\glsplural` except that the first letter is converted to uppercase.

`\Glsplural`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsplural}{\Glsplural}
```

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

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

Read in the final optional argument:

```
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentryplural{#2}#3}%
}
```

`\GLSplural` behaves like `\glsplural` except that the text is converted to uppercase.

`\GLSplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@Glsentryplural{#2}#3}%
}
```

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

`\glsfirstplural`

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

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

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

Read in the final optional argument:

```
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentryfirstplural{#2}#3}%
}
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the first letter is converted to uppercase.

`\Glsfirstplural`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
```

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

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

Read in the final optional argument:

```
\def\@Glsfirstplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
```

`\Glsfirstplural` behaves like `\glsfirstplural` except that the link text is converted to uppercase.

`\Glsfirstplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSfirstplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryfirstplural{#2}#3}%
}
```

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

`\glsname`

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

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

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

Read in the final optional argument:

```
\def\@glsname@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
```

`\Glsname` behaves like `\glsname` except that the first letter is converted to uppercase.

`\Glsname`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsname}{\Glsname}
```

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

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

Read in the final optional argument:

```
\def\@Glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
```

`\GLSname` behaves like `\glsname` except that the link text is converted to uppercase.

`\GLSname`

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

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

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

Read in the final optional argument:

```
\def\@GLSname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryname{#2}#3}}%
}
```

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

`\glsdesc`

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

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

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

Read in the final optional argument:

```
\def\@glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
```

`\Glsdesc` behaves like `\glsdesc` except that the first letter is converted to uppercase.

`\Glsdesc`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsdesc}{\Glsdesc}
```

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

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

Read in the final optional argument:

```
\def\@Glsdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentrydesc{#2}#3}%
}
```

\GLSdesc behaves like \glsdesc except that the link text is converted to uppercase.

\GLSdesc

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

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

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

Read in the final optional argument:

```
\def\@GLSdesc@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@gls@supercase{\@Glsentrydesc{#2}#3}}%
}
```

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

\glsdescplural

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

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

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

Read in the final optional argument:

```
\def\@glsdescplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@Glsentrydescplural{#2}#3}%
}
```

\Glsdescplural behaves like \glsdescplural except that the first letter is converted to uppercase.

\Glsdescplural

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsdescplural}{\Glsdescplural}
```

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

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

Read in the final optional argument:

```
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
```

\GLSdescplural behaves like \glsdescplural except that the link text is converted to uppercase.

\GLSdescplural

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

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

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

Read in the final optional argument:

```
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentrydescplural{#2}#3}%
}
```

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

\glssymbol

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

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

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

Read in the final optional argument:

```
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
```

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

\Glsymbol

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

Add sentence case mapping.

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

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

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

Read in the final optional argument:

```
\def\@Glssymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbol{#2}#3}%
}
```

`\GLSsymbol` behaves like `\glsymbol` 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}{\glssuppercase{\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*{\@glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@glsymbolplural@{#1}{#2}}{\@glsymbolplural@{#1}{#2} []}]
```

Read in the final optional argument:

```
\def\@glsymbolplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glstentrysymbolplural{#2}#3}%
}
```

`\Glsymbolplural` behaves like `\glsymbolplural` except that the first letter is converted to uppercase.

`\Glsymbolplural`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsymbolplural}{\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 `\glsymbolplural` except that the link text is converted to uppercase.

`\GLSsymbolplural`

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

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

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

Read in the final optional argument:

```
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssupercase{\glstentrysymbolplural{#2}#3}}%
}
```

`\glsuseri` behaves like `\gls` except it always uses the value given by the `user1` key and it doesn't mark the entry as used.

`\glsuseri`

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

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

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

Read in the final optional argument:

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

`\Glsuseri` behaves like `\glsuseri` except that the first letter is converted to uppercase.

`\Glsuseri`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseri}{\Glsuseri}
```


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

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

Read in the final optional argument:

```
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@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*{\@Glsuseriiv}[2] [] {%
  \new@ifnextchar[{\@Glsuseriiv@{#1}{#2}}{\@Glsuseriiv@{#1}{#2} []}]}
```

Read in the final optional argument:

```
\def\@Glsuseriiv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsuppercase{\Glsentryuseriiv{#2}#3}%
}
```

\glsuseriiv behaves like \gls except it always uses the value given by the user4 key and it doesn't mark the entry as used.

\glsuseriiv

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

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

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

Read in the final optional argument:

```
\def\@glsuseriiv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryuseriiv{#2}#3}%
}
```

\Glsuseriiv behaves like \glsuseriiv except that the first letter is converted to uppercase.

\Glsuseriiv

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuseriiv}{\Glsuseriiv}
```

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

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

Read in the final optional argument:

```
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsentryuseriv{#2}#3}%
}
```

`\Glsuseriv` behaves like `\glsuseriv` except that the link text is converted to uppercase.

`\Glsuseriv`

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

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

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

Read in the final optional argument:

```
\def\@Glsuseriv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuseriv{#2}#3}%
}
```

`\glsuserv` behaves like `\gls` except it always uses the value given by the `user5` key and it doesn't mark the entry as used.

`\glsuserv`

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

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

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

Read in the final optional argument:

```
\def\@glsuserv@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsentryuserv{#2}#3}%
}
```

`\Glsuserv` behaves like `\glsuserv` except that the first letter is converted to uppercase.

`\Glsuserv`

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuserv}{\Glsuserv}
```

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

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

Read in the final optional argument:

```
\def\@GLsuserv@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\@glsentryuserv{#2}#3}%
}
```

\GLSuserv behaves like \glsuserv except that the link text is converted to uppercase.

\GLSuserv

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

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

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

Read in the final optional argument:

```
\def\@GLSuserv@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuserv{#2}#3}%
}
```

\glsuservi behaves like \gls except it always uses the value given by the user6 key and it doesn't mark the entry as used.

\glsuservi

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

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

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

Read in the final optional argument:

```
\def\@glsuservi@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\@glsentryuservi{#2}#3}%
}
```

\Glsuservi behaves like \glsuservi except that the first letter is converted to uppercase.

\Glsuservi

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

Add sentence case mapping.

```
\glsmfuaddmap{\glsuservi}{\Glsuservi}
```

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

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

Read in the final optional argument:

```
\def\@Glsuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsentryuservi{#2}#3}%
}
```

\Glsuservi behaves like \glsuservi except that the link text is converted to uppercase.

\Glsuservi

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

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

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

Read in the final optional argument:

```
\def\@Glsuservi@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\@glsuppercase{\@glsentryuservi{#2}#3}%
}
```

Now deal with acronym related keys. First the short form:

\acrshort

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

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

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

Read in the final optional argument:

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

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

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

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

\glspostlinkhook
}
```

\Acrshort

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

Add sentence case mapping.

```
\glsmfuaddmap{\Acrshort}{\Acrshort}
```

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

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

Read in the final optional argument:

```
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%

    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

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

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

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

\glspostlinkhook
}
```

\ACRshort

```
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
```

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

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

Read in the final optional argument:

```
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsuppercase{\acronymfont{\glstryshort{#2}}#3}%
    }%
  }
```

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

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

```
\glspostlinkhook
}
```

Short plural:

\acrshortpl

```
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\@ns@acrshortpl}
```

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

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

Read in the final optional argument:

```
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

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

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

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



```

\glspostlinkhook
}

\Acrshortpl
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}

Add sentence case mapping.
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
Define the un-starred form. Need to determine if there is a final optional argument
\newcommand*{\ns@Acrshortpl}[2][{}]{%
\new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]}%
}

Read in the final optional argument:
\def\@Acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

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

Call \@gl@link Note that \@gl@link sets \glstype.
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\ACRshortpl
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}

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

Read in the final optional argument:
\def\@ACRshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%

```

```
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%
\let\gl@ifplural\@firstoftwo
\let\gl@scapscase\@thirdofthree
\let\gl@insert\@empty
\def\glscustomtext{%
  \gl@supercase{\acronymfont{\gl@entryshortpl{#2}}#3}%
}%
```

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

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

```
\gl@postlinkhook
}
```

\acrlong

```
\newrobustcmd*{\acrlong}{\@gl@hyp@opt\@ns@acrlong}
```

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

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

Read in the final optional argument:

```
\def\@acrlong#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper

    \def\glslabel{#2}%
    \let\gl@ifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \let\gl@insert\@empty
```

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

```
\def\glscustomtext{%
  \gl@entrylong{#2}#3%
}%
```

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

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

```
\gl@postlinkhook
}
```

`\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][\@ns@Acrlong]{%  
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%  
}
```

Read in the final optional argument:

```
\def\@Acrlong#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%  
  
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper  
  
    \def\glslabel{#2}%  
    \let\glsifplural\@secondoftwo  
    \let\glsupcase\@secondofthree  
    \let\glsinsert\@empty
```

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

```
\def\glscustomtext{%  
  \Glsentrylong{#2}#3%  
}%
```

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

```
\@gls@link[#1]{#2}{\csname gls@#1\glsstyle @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][\@ns@ACRlong]{%  
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%  
}
```

Read in the final optional argument:

```
\def\@ACRlong#1#2[#3]{%  
  \glsdoifexists{#2}%  
  {%
```

```
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%
\let\gl@ifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\gl@insert\@empty
```

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

```
\def\glscustomtext{%
\gl@supercase{\gl@entrylong{#2}#3}%
}%
```

Call `\@gl@link`. Note that `\@gl@link` sets `\glstyle`.

```
\@gl@link{#1}{#2}{\csname gls@glstyle @entryfmt\endcsname}%
}%
```

```
\gl@postlinkhook
}
```

Short plural:

```
\acrlongpl
```

```
\newrobustcmd*{\acrlongpl}{\@gl@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]{%
\gl@doifexists{#2}%
{%
```

```
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
```

```
\def\glslabel{#2}%
\let\gl@ifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\gl@insert\@empty
```

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

```
\def\glscustomtext{%
\gl@entrylongpl{#2}#3%
}%
```

Call `\@gl@link`. Note that `\@gl@link` sets `\glstyle`.

```
\@gl@link{#1}{#2}{\csname gls@glstyle @entryfmt\endcsname}%
}%
```

```

\glspostlinkhook
}

\Acrlongpl
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}

Add sentence case mapping.
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
Define the un-starred form. Need to determine if there is a final optional argument
\newcommand*{\ns@Acrlongpl}[2][\%
\new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}

Read in the final optional argument:
\def\@Acrlongpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty

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

Call \@gls@link. Note that \@gls@link sets \glstype.
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\ACRlongpl
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}

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

```

Read in the final optional argument:

```
\def\ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

    \let\do@gl@link@checkfirsthyper\@gl@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{%
  \glssupercase{\glsentrylongpl{#2}#3}%
}%
```

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

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

\glspostlinkhook
}
```

2.11.2 Displaying entry details without adding information to the glossary

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

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

```
\@gl@entry@field{<label>}{<field>}
```

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

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

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

`\glsletentryfield`

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

`\@Gls@entry@field` Generic first letter uppercase version.

```
\@Gls@entry@field{\label}{\field}
```

Check for PDF bookmarks performed by `\@Glsentryfield`.

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

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

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

`\@Glsentryfield`

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

`\glstexpdfstring` 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{\glstexpdfstring}{\texorpdfstring}}
{\newcommand{\glstexpdfstring}[2]{#1}}
```

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

`\glsentryname`

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

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

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

    \glsmfuexcl{\Glsentryname}
```

`\@Gls@entryname` This is a workaround in the event that the user defies the warning in the manual about not using `\Glsname` or `\Glsentryname` with acronyms. The check for PDF bookmarks is in `\Glsentryname`. First the default behaviour:

```
\newcommand*{\@Gls@entryname}[1]{%
    \@Gls@entry@field{#1}{name}%
}
```

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

```
\newrobustcmd*{\@Gls@acrentryname}[1]{%
    \ifglshaslong{#1}%
    {%
```

Just need to ensure that `\acrnameformat` is expanded, which may cause problems as non-robust semantic commands may expand to content that interferes with sentence-casing.

```
        \protected@edef\@glo@text{\@gls@entry@field{#1}{name}}%
        \expandafter\glssentencecase\expandafter{\@glo@text}%
    }%
    {%
```

Not an acronym

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

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

```
\newcommand*{\@org@Gls@acrentryname}[1]{%
    \ifglshaslong{#1}%
    {%
        \letcs\@glo@text{glo\@glsdetoklabel{#1}@name}%
    }
```


`\@gls@getbody` is defined by `mfirstuc` (which used to be part of `glossaries`).

```

\expandafter\@gls@getbody\@glo@text{}\@nil
\expandafter\ifx\@gls@body\glentrylong\relax
\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax

```

Temporarily make `\glentryshort` behave like `\Glsentryshort`. (This is on the assumption that the argument of `\acronymfont` is `\glentryshort {<label>}`, as that's the behaviour of the predefined acronym styles.) This is scoped to localise the effect of the assignment.

```

{
  \let\glentryshort\Glsentryshort
  \@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{
  \let\glentryshort\Glsentryshort
  \@glo@text
}%
\else
\expandafter\glssentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
}%
{

```

Not an acronym

```

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

```

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

```

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

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

```

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 sanitized, so unexpected results may occur if the `sort` key contained commands. This is likewise not intended to produce text in the document, other than for debugging purposes.

`\glsentrysort`

```

\newcommand*{\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}

```

`\glsentryparent` Expands to the label of the entry's parent. This expands to a label so isn't intended to produce text in the document.

```

\newcommand*{\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}

```

`\glsentryuseri` Get the first user key (as specified by the `user1` when the entry was defined).
The argument is the label associated with the entry.

```

\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}

```

`\Glsentryuseri`

```

\newcommand*{\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}

\glsmfuexcl{\Glsentryuseri}

```

`\glsentryuserii` Get the second user key (as specified by the `user2` when the entry was defined).
The argument is the label associated with the entry.

```

\newcommand*{\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}

```

`\Glsentryuserii`

```

\newcommand*{\Glsentryuserii}[1]{%
  \@Glsentryfield{#1}{userii}%
}

\glsmfuexcl{\Glsentryuserii}

```

`\glsentryuseriii` Get the third user key (as specified by the `user3` when the entry was defined).
The argument is the label associated with the entry.

```

\newcommand*{\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}

```

`\Glsentryuseriii`

```

\newcommand*{\Glsentryuseriii}[1]{%
  \@Glsentryfield{#1}{useriii}%
}

```

Add sentence case mapping.

```

\glsmfuexcl{\Glsentryuseriii}

```

`\glsentryuseriv` Get the fourth user key (as specified by the `user4` when the entry was defined).
The argument is the label associated with the entry.

```

\newcommand*{\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}

```

```

\Glsentryuseriv
    \newcommand*{\Glsentryuseriv}[1]{%
        \@Glsentryfield{#1}{useriv}%
    }

    \glsmfuexcl{\Glsentryuseriv}

\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}%
    }

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

    \glsmfuexcl{\Glsentryuserv}

\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}%
    }

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

    \glsmfuexcl{\Glsentryuservi}

\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}

\glentryshortpl Get the short plural key (as specified by the shortplural the entry was defined).
The argument is the label associated with the entry.
    \newcommand*{\glentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}

```

```

\Glsentryshortpl
    \newcommand*{\Glsentryshortpl}[1]{%
        \@Glsentryfield{#1}{shortpl}%
    }

    \glsmfuexcl{\Glsentryshortpl}

\glsentrylong Get the long key (as specified by the long the entry was defined). The argument
is the label associated with the entry.
    \newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}

\Glsentrylong
    \newcommand*{\Glsentrylong}[1]{%
        \@Glsentryfield{#1}{long}%
    }

    \glsmfuexcl{\Glsentrylong}

\glsentrylongpl Get the long plural key (as specified by the longplural the entry was defined).
The argument is the label associated with the entry.
    \newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}

\Glsentrylongpl
    \newcommand*{\Glsentrylongpl}[1]{%
        \@Glsentryfield{#1}{longpl}%
    }

    \glsmfuexcl{\Glsentrylongpl}

    Short cut macros to access full form:

\glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*{\glsentryfull}[1]{%
        \glsentrylong{#1} (\glsentryshort{#1})%
    }

\Glsentryfull This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*{\Glsentryfull}[1]{%
        \Glsentrylong{#1} (\glsentryshort{#1})%
    }

    \glsmfuexcl{\Glsentryfull}

\glsentryfullpl This definition shouldn't actually be used. This command will be redefined by
the acronym style.
    \newcommand*{\glsentryfullpl}[1]{%
        \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
    }

```

`\Glsentryfullpl` This definition shouldn't actually be used. This command will be redefined by the acronym style.

```
\newcommand*\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\Glsentryshortpl{#1})%
}%
%\changes{4.54}{2024-04-03}{change map to exclusion}

\glsmfuexcl{\Glsentryfullpl}
```

`\glsentrynumberlist` Displays the number list as is.

```
\newcommand*\glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
```

`\glsdisplaynumberlist` Formats the number list for the given entry label. Doesn't work with hyperref.

```
\@ifpackageloaded{hyperref} {%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
  \newcommand*\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup

      \protected@edef\@glo@label{\glsdetoklabel{#1}}%
      \let\@org@glnumberformat\glnumberformat
      \def\glnumberformat##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[\langle options \rangle]{\langle label \rangle}
```

Add a term to the glossary without generating any link text. The optional argument indicates which counter to use, and how to format it (using a key-value list) the second argument is the entry label. Note that *\langle options \rangle* only has two keys: `counter` and `format` (the `types` key will be ignored).

`\glsadd`

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

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

```
\@gls@adjustmode
```

```
\glsdoifexists{#2}%
```

```
{%
```

```
\def\@glsnumberformat{glsnumberformat}%
```

```
\protected@edef\@gls@counter{\csname glo@\glsdetoklabel{#2}@counter\endcsname}%
```

```
\setkeys{glossadd}{#1}%
```

Store the entry's counter in `\theglsentrycounter`

```
\@gls@saveentrycounter
```

Define sort key if necessary:

```
\@gls@setsort{#2}%
```

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

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

```
}%
```

```
}
```

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

```
\glsmfuexcl{\glsadd}
```

`\@gls@adjustmode`

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

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

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

`\glsaddall[option list]`

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

`\glsaddall`

```
\newrobustcmd*{\glsaddall}[1] [] {%
```

```
\protected@edef\@glo@type{\@glo@types}%
```

```
\setkeys{glossadd}{#1}%
```

```
\forallglsentries[\@glo@type]{\@glo@entry}{%
```

```
\glsadd{#1}{\@glo@entry}%
```

```
}%
```

```
}
```

```
\glsaddallunused[<glossary types>]
```

`\glsaddallunused`

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

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

`\glsignore`

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

2.13 Creating associated files

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

The special characters are stored in `\@gls@actualchar`, `\@gls@encapchar`, `\@gls@levelchar` and `\@gls@quotechar` to make them easier to use later, but don't change these values, because the characters are encoded in the command definitions that are used to escape the special characters (which means that the user no longer needs to worry about `makeindex` special characters).

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

`\glsopenbrace` Define `\glsopenbrace` to make it easier to write an opening brace to a file.

```
\edef\glsopenbrace{\expandafter\@gobble\string\{}
```

`\glsclosebrace` Define `\glsclosebrace` to make it easier to write an opening brace to a file.

```
\edef\glsclosebrace{\expandafter\@gobble\string\}}
```

`\glsbackslash` Define `\glsbackslash` to make it easier to write a backslash to a file.

```
\edef\glsbackslash{\expandafter\@gobble\string\}
```

`\glsquote` Define command that makes it easier to write quote marks to a file in the event that the double quote character has been made active.

```
\edef\glsquote#1{\string"#1\string"}
```

`\glpercentchar` Define `\glpercentchar` to make it easier to write a percent character to a file.

```
\edef\glpercentchar{\expandafter\@gobble\string\%}
```

`\glstildechar` Define `\glstildechar` to make it easier to write a tilde character to a file.

```
\edef\glstildechar{\string~}
```

`\@glsfirstletter` Define the first letter to come after the digits 0,...,9. Only required for xindy.

```
\ifglsxindy
```

```
\newcommand*{\@glsfirstletter}{A}
```

```
\fi
```

`\GlsSetXdyFirstLetterAfterDigits` Sets the first letter to come after the digits 0,...,9. The starred version sanitizes.

```
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
```

```
\@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
```

```
\ifglsxindy
```

```
\newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
```

```
\renewcommand*{\@glsfirstletter}{#1}}
```

```
\newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
```

```
\renewcommand*{\@glsfirstletter}{#1}%
```

```
\@onelevel@sanitize\@glsfirstletter
```

```
}
```

```
\else
```

```
\newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
```

```
\glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}
```

```
\newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
```

```
\@GlsSetXdyFirstLetterAfterDigits
```

```
}
```

```
\fi
```

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

```
\ifglsxindy
```

```
\newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
```

```
\fi
```

`\GlsSetXdyNumberGroupOrder` Sets the relative location of the number group. The starred version sanitizes.

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

```
\@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
```

```
}
```

```
\ifglsxindy
```

```
\newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
```

```
\renewcommand*{\@xdynumbergrouporder}{#1}%
```

```
}
```

```
\newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
```

```
\renewcommand*{\@xdynumbergrouporder}{#1}%
```

```
\@onelevel@sanitize\@xdynumbergrouporder
```

```
}
```

```
\else
```

```
\newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
```

```
\glsnxindywarning\GlsSetXdyNumberGroupOrder}
```

```

\newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
  \@GlsSetXdyNumberGroupOrder}
\fi

\@glsmminrange Define the minimum number of successive location references to merge into a
range.
\newcommand*{\@glsmminrange}{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*{\@glsmminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \glsnnoxindywarning\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
    \@glscaddpredefinedattributes
Open the file.
    \openout\glswrite=\istfilename
Write header comment at the start of the file
    \write\glswrite{;; xindy style file created by the glossaries
      package}%
    \write\glswrite{;; for document '\jobname' on
      \the\year-\the\month-\the\day}%
Specify the required styles
    \write\glswrite{^^J; required styles^^J}
    \@for\@xdystyle:=\@xdyrequiredstyles\do{%
      \ifx\@xdystyle\@empty
      \else
        \protected@write\glswrite{{(require
          \string"\@xdystyle.xdy\string")}}%
      \fi
    }%
  }%

```

List the allowed attributes (possible values used by the format key)

```
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
```

Define any additional alphabets

```
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
```

Define location classes.

```
\write\glswrite{^^J; location class definitions^^J}%
```

As from version 3.0, locations are now specified as $\{\langle Hprefix \rangle\}\{\langle number \rangle\}$, so need to add all possible combinations of location types.

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

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

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

Nested iteration over all classes:

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

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

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

```
\write\glswrite{\@xdyuserlocationdefs}%
```

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

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

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

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

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

```
\@xdycrossrefhook
```

List the order to sort the classes.

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

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

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

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

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

```
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{ }\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
```

```
}%
}%
```

Add the end part of the open tag and the rest of the markup-index information:

```
\write\glswrite{%
  \string\begin
    {theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
    \space\space:close \string"\glpercentchar\glstildechar n\string
    \end{theglossary}\string\glossarypostamble
    \glstildechar n\string" ^^J\space\space\space
  :tree)}%
```

Specify what to put between letter groups

```
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\glstildechar n\string")}%
```

Specify what to put between entries

```
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glsresetentrylist
  \glstildechar n\string")}%
```

Specify how to format entries

```
\write\glswrite{(markup-locclass-list :open
  \string"\glsopenbrace\string\glossaryentrynumbers
  \glsopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
```

Specify how to separate location numbers

```
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
```

Specify how to indicate location ranges

```
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
```

Specify 2-page and 3-page suffixes, if defined. First, the values must be sanitized to write them explicitly.

```
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF

\ifx\gls@suffixF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
```


Specify how to format locations.

```
\write\glswrite{^^J; define format to use for locations^^J}%  
\write\glswrite{\@xdylocref}%
```

Specify how to separate letter groups.

```
\write\glswrite{^^J; define letter group list format^^J}%  
\write\glswrite{(markup-letter-group-list  
:sep \string"\string\glsgroupskip\glstildechar n\string")}%
```

Define letter group headings.

```
\write\glswrite{^^J; letter group headings^^J}%  
\write\glswrite{(markup-letter-group  
:open-head \string"\string\glsgroupheading  
\glsoopenbrace\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{\glspcrcentchar\space makeindex style file  
created by the glossaries package}  
\write\glswrite{\glspcrcentchar\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

```

Hook for any additional information:

```
\@gls@writeisthook
```

Close the file and disable \writeist.

```
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
```

\@gls@writeisthook Hook used just before the style file is closed:

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

\GlsSetWriteIstHook Set the hook.

```
\newcommand*\GlsSetWriteIstHook[1]{\renewcommand*\@gls@writeisthook{#1}}
\@onlypremakeg\GlsSetWriteIstHook
```

\GlsAppToWriteIstHook Append to the hook.

```
\newcommand*\GlsAppToWriteIstHook[1]{\appto\@gls@writeisthook{#1}}
\@onlypremakeg\GlsAppToWriteIstHook
```

\@@gls@extramakeindexopts

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

\GlsAddToMakeIndexOpts

```
\ifglxindy
\newcommand*\GlsAddToMakeIndexOpts[1]{\glsnomakeindexwarning\GlsAddToMakeIndexOpts}
\else
\newcommand*\GlsAddToMakeIndexOpts[1]{%
\appto\@@gls@extramakeindexopts{ #1}%
}
\fi
```

\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}%
{%
\ifdefstring\@gls@makeindexapp{makeindex}
{%

```

```

\GlsAddToMakeIndexOpts{-g}%
}%
{}%
}%
{}%
}%
{}%

```

Need to redefine \@gls@checkquote

```

\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%

```

```

\noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@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.

\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}}%
\@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{-l }}%
{\let\@gls@order\empty}%
\@gls@runshell{\@gls@makeindexapp\space \@gls@order
\@gls@extramakeindexopts

```



```

-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.

```

\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%

```

Set `\@gls@langmod`:

```

\gls@xdy@set@lang{#1}%

\ifdefvoid\gls@codepage
{\let\@gls@codepage\@empty}%
{\def\@gls@codepage{-C \gls@codepage\space}}%
\@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{-l }}%
        {\let\@gls@order\@empty}%
        \@gls@runshell{\@gls@makeindexapp\space \@gls@order
          \@gls@extramakeindexopts
          -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
}

```

`\@glo@assign@sortkey`

```

\newcommand{\@glo@assign@sortkey}{\@glo@no@assign@sortkey}

```

`\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
    \let\glsindexingsetting\@gls@makeindexapp
  \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 set, write it:

```

\ifdefempty\@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}}

```

Write the name of the makeindex or makeindex-substitute application.

```
\ifglxindy
\else
  \protected@write\@auxout{}\string\providecommand\string\@glsmakeindexappname[1]{}
  \protected@write\@auxout{}\string\@glsmakeindexappname{\@gls@makeindexapp}%
\fi
```

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@doddeflistparser{\noexpand\DeclareListParser
  {\noexpand\glsnumlistparser}{\delimN}}%
\@gls@doddeflistparser
\fi
```

Prevent user from also using `\makenoidxglossaries`

```
\let\makenoidxglossaries\@no@makeglossaries
```

Prohibit sort key in printgloss family:

```
\renewcommand*\@printgloss@setsort{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
```

Check the automake setting:

```
\ifglautomake
\renewcommand*{\@gls@doautomake}{%
\protected@edef\gls@tmp{\@gls@automake@types}%
\@for\@gls@type:=\gls@tmp\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}%
}%
}%
\fi
```

Check the sort setting:

```
\@glo@check@sortallowed\makeglossaries
}%
}
```

Must occur in the preamble:

```
\@onlypreamble{\makeglossaries}
```

`ssaries@warn@noprintglossary`

```
\newcommand\@makeglossaries@warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
{%
\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
```

`\glswrite` The definition of `\glswrite` has now been moved to `\makeglossaries` so that it's only defined if needed.

If `\makeglossaries` hasn't been used, issue a warning. Also issue a warning if neither `\printglossaries` nor `\printglossary` have been used.

```
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
```

`\makenoidxglossaries` Analogous to `\makeglossaries` this activates the commands needed for `\printnoidxglossary`

```
\newcommand*{\makenoidxglossaries}{%
\def\glsindexingsetting{noidx}%
\@domakeglossaries
}%
```

Redefine empty glossary warning:

```
\renewcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{Empty glossary for
    \string\printnoidxglossary[type={##1}].
    Rerun may be required (or you may have forgotten to use
    commands like \string\gls)}%
}%
```

Don't escape makeindex/xindy characters:

```
\let\@gls@checkmkidxchars\@gobble
```

Don't escape locations:

```
\glsesclocationsfalse
```

Write glossary information to aux instead of glossary files

```
\let\@do@wrglossary\gls@noidxglossary
```

Switch on group headings that use the character code:

```
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
```

Allow see key:

```
\let\gls@checkseeallowed\relax
```

Redefine cross-referencing macro:

```
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef\@gls@label{\glsdetoklabel{##1}}%
  \protected@write\@auxout{}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
    {%
      \string\glsseeformat##2}%
    }%
  }%
  \@gls@noidx@addtorefs{\@gls@label}%
}%
```

If user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
```

Change warning about no glossaries

```
\def\warn@noprntglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
```

Suppress warning about no \makeglossaries

```
\let\warn@nomakeglossaries\relax
```

Prevent user from also using \makeglossaries

```
\let\makeglossaries\@no@makeglossaries
```

Allow sort key in printgloss family:

```
\renewcommand*{\@printgloss@setsort}{%  
  \let\@glo@assign@sortkey\@glo@assign@sortkey
```

Initialise default sort order:

```
\def\@glo@sorttype{\@glo@default@sorttype}%  
}%
```

All entries must be defined in the preamble:

```
\renewcommand*\new@glossaryentry[2]{%  
  \PackageError{glossaries}{Glossary entries must be  
    defined in the preamble^^Jwhen you use  
    \string\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  
  {%  
    \glsnoidxloclist{\@gls@loclist}%  
  }%  
  {%  
    ??\glsdoifexists{##1}%  
    {%  
      \GlossariesWarning{Missing location list for ‘##1’. Either  
        a rerun is required or you haven’t referenced the entry}%  
    }%  
  }%  
}%
```

Redefine \glsdisplaynumberlist

```
\renewcommand*{\glsdisplaynumberlist}[1]{%  
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%  
  \ifdef\@gls@loclist  
  {%  
    \def\@gls@noidxloclist@sep{%  
      \def\@gls@noidxloclist@sep{%  
        \def\@gls@noidxloclist@sep{%  
          \glsnumlistsep  
        }%  
      \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%  
    }%  
  }%  
}%
```

```

\def\@gls@noidxloclist@finalsep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
\@gls@noidxloclist@finalsep
\@gls@noidxloclist@prev
}%
{%
??\glsdoifexists{##1}%
{%
\GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven’t referenced the entry}%
}%
}%
}%

```

Provide a generic way of iterating through the number list:

```

\renewcommand*{\glsnumberlistloop}[3]{%
\letcs{\@gls@loclist}{\glo\@glsdetoklabel{##1}@loclist}%
\let\@gls@org\@glsnoidxdisplayloc\glsnoidxdisplayloc
\let\@gls@org\@glsseeformat\glsseeformat
\let\glsnoidxdisplayloc##2\relax
\let\glsseeformat##3\relax
\ifdef\@gls@loclist
{%
\forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
}%
{%
??\glsdoifexists{##1}%
{%
\GlossariesWarning{Missing location list for ‘##1’. Either
a rerun is required or you haven’t referenced the entry}%
}%
}%
\let\glsnoidxdisplayloc\@gls@org\@glsnoidxdisplayloc
\let\glsseeformat\@gls@org\@glsseeformat
}%

```

Modify sanitize sort function

```

\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort

```

Check sort option allowed.

```

\@glo@check@sortallowed\makenoidxglossaries

```

Implement nonumberlist

```

\let\@gls@savenonumberlist\gls@noidx@nonumberlist

```

Do rerun check at the end of the document.

```

\AtEndDocument{\GlsNoIdxDoRerunCheck}%
}%
}

```


Preamble-only command:

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

`\glsnumberlistloop`

```
\glsnumberlistloop{<label>}{<handler>}{<xr handler cs>}
```

```
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
```

`\glsnoidxnumberlistloophandler` Handler macro for `\glsnumberlistloop`. (The argument should be in the form

```
\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<n>}
```

```
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
```

`\@no@makeglossaries` Can't use both `\makeglossaries` and `\makenoidxglossaries`

```
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}
```

`\@gls@noref@warn` Warning when no instances of `\@gls@reference` found.

```
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
```

2.14 Writing information to associated files

`\gls@noidxglossary` Write the glossary information to the aux file (for the 'noidx' method).

```
\newcommand*{\gls@noidxglossary}{%
  \protected@write\auxout{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}}%
      }%
  }%
  \@gls@noidx@addtorefs{\@gls@label}%
}
```

\ExplSyntaxOn

\@gls@noidx@addtorefs

```
\newcommand\@gls@noidx@addtorefs[1]{%
\exp_args:Nc \__glossaries_noidx_add_to_noidx_refs:n { #1 }
}
```

```
\cs_new:Nn \__glossaries_noidx_add_to_noidx_refs:n
{
\seq_if_in:NnF \g__glossaries_noidx_refs_seq { #1 }
{
\seq_gput_right:Nn \g__glossaries_noidx_refs_seq { #1 }
}
```

If entry has a parent, make sure that is also added.

```
\tl_if_empty:cF { glo@ #1 @parent }
{
\exp_args:Nv \__glossaries_noidx_add_to_noidx_refs:n
{ glo@ #1 @parent }
}
}
```

Keep track of entries that have been listed with \printnoidxglossary.

```
\seq_new:N \g__glossaries_noidx_listed_seq
\cs_new:Nn \__glossaries_noidx_add_to_listed:n
{
\seq_if_in:NnF \g__glossaries_noidx_listed_seq { #1 }
{
\seq_gput_right:Nn \g__glossaries_noidx_listed_seq { #1 }
}
}
```

Keep track of entries that have been referenced.

```
\seq_new:N \g__glossaries_noidx_refs_seq
\cs_new:Nn \__glossaries_noidx_rerun_warn:
{
\GlossariesWarning
{
Rerun ~ required ~ to ~ get ~
\token_to_str:N \printnoidxglossary \c_space_tl up ~ to ~ date
}
\cs_set_eq:NN \__glossaries_noidx_rerun: \relax
}
\cs_new:Nn \__glossaries_noidx_rerun: { }
\cs_new:Nn \__glossaries_noidx_check_refs:
{
\seq_map_inline:Nn \g__glossaries_noidx_refs_seq
{
\__glossaries_if_has_glsref_or_parent_of:nF { ##1 }
{
\GlossariesWarning
}
```

```

    {
      entry ~ '##1' ~ indexing ~ changed ~ since ~
        previous ~ run ~ (or ~ no ~ previous ~ run)
    }
    \cs_set_eq:NN
      \__glossaries_noidx_rerun:
      \__glossaries_noidx_rerun_warn:
  }
}
\seq_map_inline:Nn \g__glossaries_noidx_listed_seq
{
  \seq_if_in:NnF \g__glossaries_noidx_refs_seq { ##1 }
  {
    \GlossariesWarning
    {
      entry ~ '##1' ~ listed ~ in ~ glossary ~ but ~
        not ~ indexed ~ this ~ run
    }
    \cs_set_eq:NN
      \__glossaries_noidx_rerun:
      \__glossaries_noidx_rerun_warn:
  }
}
\__glossaries_noidx_rerun:
}
\NewDocumentCommand \GlsNoIdxDoRerunCheck { }
{
  \__glossaries_noidx_check_refs:
  \cs_set_eq:NN \GlsNoIdxDoRerunCheck \relax
}
\ExplSyntaxOff

```

`\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.

```

  \foralllglossaries{\@glo@type}{%

```

Check for empty glossaries (patch provided by Patrick Häcker)

```

    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%

```

```

{%
  \edef\gls@tmp{\expandafter\the
    \csname glo@\@glo@type @filetok\endcsname}%
}%
\ifx\gls@tmp\@empty
  \ifx\@glo@type\glsdefaulttype
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries.^^JRemember to use package option ‘nomain’ if
you
      don’t want to^^Juse the main glossary}%
  \else
    \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
      entries}%
  \fi
\else
  \@glsopenfile{\glswrite}{\@glo@type}%
  \immediate\write\glswrite{%
    \expandafter\the
      \csname glo@\@glo@type @filetok\endcsname}%
  \immediate\closeout\glswrite
  \fi
}%
}

```

As from v4.10, the `\glossary` command isn’t used by the `glossaries` package. Since the user isn’t expected to use this command (as `glossaries` takes care of the particular format required for `makeindex/xindy`) there’s no need for a user level command. Using a custom internal command prevents any conflict with other packages (and with the `\mark` mechanism).

The associated number should be stored in `\theglsentrycounter` before using `\gls@glossary`.

```

\gls@glossary
  \newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
  }

```

`\@gls@glossary`

`\@gls@glossary{<type>}{<indexing info>}`

(In v4.10, `\glossary` was redefined to `\@gls@glossary` to avoid conflict with other packages.) Initially define internal `\@gls@glossary` to ignore its argument. Indexing will be enabled when `\@gls@glossary` is redefined by `\@makeglossary`.

This command was originally defined to do `\@index{<indexing info>}` so that it behaved much like `\index`. The definition was then changed to use `\index` as `memoir` changes the definition of `\@index`. (Thanks to Dan Luecking for pointing this out.)

However, if normal indexing is enabled (for example with `\makeindex`) but no glossary lists are required (so `\makeglossary` isn't used), then `\index` will cause a problem here. The `\@index` trick allows for special characters within *<indexing info>* (so you can do, for example, `\index{%@%}`), and the original design of `\@glossary` here was actually a legacy from the old `glossary` package. With the `glossaries` package, the indexing information supplied in the second argument is more constrained and just consists of the sort value (given by the sort key), the actual value (given by `\glossentry{<label>}` or `\subglossentry{<level>}{<label>}`), and the format. This means that there's no need to worry about special characters appearing in the second argument as they can't be in the label or sort value. (If they are in the sort value then the category code would've needed to be changed when the entry was defined or `\glsperscentchar` would be needed with the sort sanitization switched off.) This means that it's safe to simply ignore the second argument.

```
\newcommand*{\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}
```

This is a convenience command to set `\@gls@glossary`. It's used by `\@makeglossary` and then redefined to do nothing, as it only needs to be done once.

`\@gls@renewglossary`

```
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begingroup\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
```

The `\gls@wrglossary` command is defined to have two arguments. The first argument is the glossary type, the second argument is the glossary entry (the format of which is set in `\glslink`).

`\gls@wrglossary`

```
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
      \expandafter{\@gls@tmp^^J}%
  \else
    \ifcsdef{glo@#1@file}%
    {%
      \expandafter\protected@write\csname glo@#1@file\endcsname{%
        \gls@disablepagerefexpansion}{#2}%
      }%
    {%
      \ifignoredglossary{#1}{}%
      {%

```

```

\GlossariesWarning{No file defined for glossary ‘#1’}%
}%
}%
\fi
\endgroup\@esphack
}

\@do@wrglossary
\newcommand*{\@do@wrglossary}[1]{%
\gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}

\glswriteentry Provide a user level command so the user can customize whether or not a line
should be added to the glossary. The arguments are the label and the code that
writes to the glossary file.
\newcommand*{\glswriteentry}[2]{%
\ifglsindexonlyfirst
\ifglsused{#1}{#2}%
\else
#2%
\fi
}

\gls@protected@pagefmts List of page formats to be protected against expansion.
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}

\gls@disablepagerefexpansion As from v4.50, this will only do the page protection formats for escla-
tions=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 ($\backslash\langle csname \rangle\c@page$ must be valid).

```

\newcommand*{\glsaddprotectedpagefmt}[1]{%
  \ifglseclocations
    \protected@eappto\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}}%
    \csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
    \protected@eappto\@wrglossarynumberhook{%
      \noexpand\let\expandonce{\csname org@gls#1\endcsname}%
      \expandonce{\csname#1\endcsname}%
      \noexpand\def\expandonce{\csname#1\endcsname}{%
        \noexpand\@wrglossary@pageformat
        \expandonce{\csname gls#1page\endcsname}%
        \expandonce{\csname org@gls#1\endcsname}%
      }%
    }%
  \else
    \GlsWarnAddProtectedPageFmt
  \fi
}

```

```
\GlsWarnAddProtectedPageFmt
```

```

\newcommand{\GlsWarnAddProtectedPageFmt}{%
  \GlossariesWarning{\string\glsaddprotectedpagefmt\space has no
    effect with esclocations=false}%
}

```

$\backslash\@wrglossarynumberhook$ Hook used by $\backslash\@do@wrglossary$

```
\newcommand*\@wrglossarynumberhook{}
```

```
\@wrglossary@pageformat
```

```

\newcommand{\@wrglossary@pageformat}[3]{%
  \ifx#3\c@page #1\else #2#3\fi
}

```

`\@@do@wrglossary` Write the glossary entry in the appropriate format.

```
\newcommand*{\@@do@wrglossary}[1]{%
  \ifglsclocations
    \@@do@esc@wrglossary{#1}%
  \else
    \@@do@noesc@wrglossary{#1}%
  \fi
}
```

`\glswrglossdisableanchorcmds`

```
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
}
```

`\glswrglosslocationtarget` Must expand. May be used to alter the location suffix in the `\glshypernumber` anchor.

```
\newcommand{\glswrglosslocationtarget}[1]{#1}
```

`\glswrglosslocationtextfmt` Used to encapsulate the hyperlink text in `\glshypernumber`.

```
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
```

`\@@do@noesc@wrglossary` Write the glossary entry in the appropriate format. The locations don't need to be pre-processed before writing the information to the glossary file, but the prefix still needs to be found.

```
\newcommand*{\@@do@noesc@wrglossary}[1]{%
```

Don't fully expand yet.

```
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
```

Find the prefix if `\@glsHlocref` and `\@glslocref` aren't the same.

```
\ifx\@glsHlocref\@glslocref
  \def\@glo@counterprefix{}%
\else
```

The value of the counter isn't important here as it's the prefix that's of interest. (`\c@page` will have the same value in both `\theglentrycounter` and `\theHglentrycounter` at this point, even if it hasn't been updated yet. The page number is not expected to occur in the prefix.)

```
\@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
```

De-tok label if required.

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

Write the information to file:

```
\@@do@wrglossary
}
```


`\ifglswrallowprimitivemods` Conditional to determine whether or not `\@@do@esc@wrglossary` should be allowed to temporarily redefine `\the` and `\number`. Version 4.50 switched the default to false.

```
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
```

`\@@do@esc@wrglossary` Write the glossary entry in the appropriate format. (Need to set `\@glsnumberformat` and `\@gls@counter` prior to use.) The argument is the entry’s label. This is far more complicated with `xindy` than with other indexing methods. There are two necessary but conflicting requirements with `xindy`:

1. all backslashes in the location must be escaped;
2. `\c@page` can’t be prematurely expanded.

(With `makeindex` there’s the remote possibility that the page compositor is a `makeindex` special character, so that would also need to be escaped.)

For example, suppose `\thepage` is defined as

```
%\renewcommand{\thepage}{\tally{page}}
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

where `\tallynum` is a robust command that takes a number as its argument. With all indexing methods other than `xindy`, a deferred write with `\thepage` as the location will expand to `\tallynum{<n>}` where `<n>` is the page number. Since the write is deferred, the page number is correct. (`makeindex` won’t accept this location format, but `\makenoidxglossaries` and `bib2gls` are quite happy with it.) Unfortunately, this fails with `xindy` because `xindy` interprets this location as `\tallynum{<n>}` because `\t` represents a the character “t”. The location must be written as `\\tallynum{<n>}`.

This means that the location `\tally{page}` must be expanded and then the backslashes must be doubled. Unfortunately `\c@page` mustn’t be expanded until the deferred write is performed, so the location actually needs to be expanded to `\tallynum{\the\c@page}` but the backslashes in `\the\c@page` mustn’t be escaped. All other backslashes must be escaped. (In this case, only the backslash in `\tallynum` but the location format may include other control sequences.) The code below works on the assumption that commands like `\tally` are defined in the form

```
%\newcommand{\tally}[1]{\tallynum{\expandafter\the\csname c@#1\endcsname}}
%
```

(note the use of `\expandafter` and `\name`) or in the form

```
%\newcommand{\tally}[1]{\tallynum{\arabic{#1}}}%
%
```

In the second case, `\arabic` is one of the known commands that's temporarily adjusted to prevent `\c@page` from being prematurely expanded. In the first case, `\the` is temporarily modified (unless `\glswrallowprimitivemodsfalse`) to check if it's followed by `\c@page`. The `\expandafter` ensures that it is. If `\tally` is defined in another way that hides `\c@page` for example using `\the\value{#1}` then the process fails.

With `makeindex`, `\tallynum` needs to expand to just the decimal number while writing the location to the glossary file, otherwise `makeindex` will reject it. This can be done by defining `\glstallypage` so that `\tally` can locally be set to `\arabic` while expansion is occurring. Again, `\c@page` must be protected from expansion until the deferred write occurs.

The expansion before the write occurs also allows the hyper prefix to be determined where `\theH{counter}` is defined in the form `\langle prefix \rangle.\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!
\beginingroup
```

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@mkidxrangechar\@glsnumberformat\@nil
\def\@glo@range{%
\expandafter\if\@glo@prefix(\relax
\def\@glo@range{:open-range}%

```

```

\else
\expandafter\if\@glo@prefix)\relax
\def\@glo@range{:close-range}%
\fi
\fi

```

Write to the glossary file using xindy syntax.

```

\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
(indexentry :tkey (\csname glo@\@gls@label @index\endcsname)

:locoref \string{"\@glo@counterprefix}{\@gls@locoref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else

```

Convert the format information into the format required for makeindex

```

\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%

```

Write to the glossary file using makeindex syntax.

```

\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locoref}}%
\fi
}

```

`\@xp@gls@getcounterprefix`

```

\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgrou
\glswrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%
\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\@do@gls@getcounterprefix
}

```

`\@gls@getcounterprefix` Get the prefix that needs to be prepended to counter in order to get the hyper counter. (For example, with the standard article class and hyperref, `\theequation` needs to be prefixed with `\section num` to get the equivalent `\theHequation`.) NB this assumes that the prefix ends with a dot, which is the standard. (Otherwise it makes the xindy location classes more complicated.)

The expansion has been moved to `\@xp@gls@getcounterprefix`.

```

\newcommand*\@gls@getcounterprefix[2]{%
\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else

```

```

\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix

Warn if no prefix can be formed.
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been^^J
referenced but does not exist”}%
\fi
\fi
}

```

2.15 Glossary Entry Cross-References

`\@do@seeglossary` Write the glossary entry with a cross reference. The first argument is the entry’s label, the second must be in the form `[\langle tag \rangle]{\langle list \rangle}`, where `\langle tag \rangle` is a tag such as “see” and `\langle list \rangle` is a list of labels.

```

\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
: tkey (\csname glo@#1@index\endcsname)
: xref (\string"\@gls@xref\string")
: attr \string"see\string"
)}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}

```

`\@gls@fixbraces` If no optional argument is specified, list needs to be enclosed in a set of braces.

```

\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces

```

```

        \else
        \def#1{{#2#3}}%
        \fi
    }

\@@gls@fixbraces

\def\@@gls@fixbraces#1[#2]#3\end@fixbraces{%
    \def#1{[#2]{#3}}%
}

\glssee \glssee{<label>}{<cross-ref list>}
\newrobustcmd*{\glssee}[3][\seename]{%
    \do@seeglossary{#2}{[#1]{#3}}%
    \newcommand*{\@glssee}[3][\seename]{%
        \glssee[#1]{#3}{#2}}
}

\glsseeformat The first argument specifies what tag to use (e.g. “see”), the second argument is
a comma-separated list of labels. The final argument (the location) is ignored.
\newrobustcmd*{\glsseeformat}[3][\seename]{%
    \emph{#1} \glsseelist{#2}}

\glsseelist \glsseelist{<list>} formats list of entry labels.
\newrobustcmd*{\glsseelist}[1]{%
    If there is only one item in the list, set the last separator to do nothing.
    \let\@gls@dolast\relax
    Don’t display separator on the first iteration of the loop
    \let\@gls@donext\relax
    Iterate through the labels
    \@for\@gls@thislabel:=#1\do{%
    Check if on last iteration of loop
        \ifx\@xfor@nextelement\@nnil
        \@gls@dolast
        \else
        \@gls@donext
        \fi
    Display the entry for this label. (Expanding label as it’s a temporary control
    sequence that’s used elsewhere.)
        \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    Update separators
        \let\@gls@dolast\glsseelastsep
        \let\@gls@donext\glsseesep
    }%
}

```

`\glsseelastsep` Separator to use between penultimate and ultimate entries in a cross-referencing list.

```
\newcommand*{\glsseelastsep}{\space\andname\space}
```

`\glsseesep` Separator to use between entries in a cross-referencing list.

```
\newcommand*{\glsseesep}{, }
```

`\glsseeitem` `\glsseeitem{<label>}` formats individual entry in a cross-referencing list.

```
\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
```

`\glsseeitemformat` Introduced in v3.0 to replace use of `\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.

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

`\@printgloss@checkexists@allowignored` For use with `\printunsrtglossary`.

```
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
\s@ifglossaryexists{#1}%
{#2}%
{\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
```

`\@printgloss@checkexists`

```
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
```

`\printglossary` has an optional argument. The default value is to set the glossary type to the main glossary.

```
\newcommand*{\printglossary}[1][type=glndefaulttype]{%
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\@printglossary{#1}{\@print@glossary}%
}
```

The `\printglossaries` command will do `\printglossary` for each glossary type that has been defined. It is better to use `\printglossaries` rather than individual `\printglossary` commands to ensure that you don't forget any new glossaries you may have created. It also makes it easier to chop and change the value of the `acronym` package option. However, if you want to list the glossaries

in a different order, or if you want to set the title or table of contents entry, or if you want to use different glossary styles for each glossary, you will need to use `\printglossary` explicitly for each glossary type.

```
\printglossaries
    \newcommand*{\printglossaries}{%
      \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
    }

\printnoidxglossary Provide an alternative to \printglossary that doesn't require an external in-
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 \@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}}%
```

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@dotoctitle
```

Set the glossary style

```
\@glossarystyle
```

Added a way to fetch the current entry label (v3.08 updated for new `\glossentry` and `\subglossentry`, but this is now only needed for backward compatibility):

```
\let\gls@org@glossaryentryfield\glossentry  
\let\gls@org@glossarysubentryfield\subglossentry  
\renewcommand{\glossentry}[1]{%  
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%  
\gls@org@glossaryentryfield{##1}%  
}%  
\renewcommand{\subglossentry}[2]{%
```

```

\protected@xdef\glscurrententrylabel{\glsetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%

\@gls@preglossaryhook

```

Now do the handler macro that deals with the actual glossary:

```

#2%
}%

```

End the current scope

```

\gls@printglossary@end
Reset \glossaryentrynumbers
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
Suppress warning about no \printglossary
\global\let\warn@noprintglossary\relax
}

```

`\gls@printglossary@begin`

```

\newcommand{\gls@printglossary@begin}{\begingroup}

```

`\gls@printglossary@end` See <https://github.com/latex3/tagging-project/issues/48>

```

\newcommand{\gls@printglossary@end}{%
\expandafter\endgroup\if@endpe\@doendpe\fi
}

```

`\@print@glossary` Internal workings of `\printglossary` dealing with reading the external file.

```

\newcommand{\@print@glossary}{%

```

Some macros may end up being expanded into internals in the glossary, so need to make @ a letter. (Unlikely to be a problem since v3.08a but kept for backward compatibility.)

```

\makeatletter

```

Input the glossary file, if it exists.

```

\input@{\jobname.\csname @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`.

```

\ifglsxindy

```

Set `\@gls@langmod`:

```

\gls@xdy@set@lang{\glo@type}%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%

```

If the user removes the glossary package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\noexpand\immediate\noexpand\write\@auxout{%
  \string\providecommand\string\@xdylanguage[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%
  \string\@xdylanguage{\@glo@type}{\@gls@langmod}}%
}%
}%
\do@auxoutstuff
\edef\@do@auxoutstuff{%
  \noexpand\AtEndDocument{%

```

If the user removes the glossaries package from their document, ensure the next run doesn't throw a load of undefined control sequence errors when the aux file is parsed.

```

\noexpand\immediate\noexpand\write\@auxout{%
  \string\providecommand\string\@gls@codepage[2]{}%
\noexpand\immediate\noexpand\write\@auxout{%
  \string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
}%
}%
\do@auxoutstuff
\fi

```

Activate warning if `\makeglossaries` hasn't been used.

```

\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}

```

The sort macros all have the syntax:

```
\@glo@sortmacro@<order>{\<type>}
```

where `<order>` is the sort order as specified by the sort key and `<type>` is the glossary type. (The referenced entry list is stored in `\@glsref@<type>`). The actual sorting is done by `\@glo@sortentries{<handler>}{<type>}`.

New to v4.55: Check if `datatool-base v3.0+` is installed, which provides the command `\datatool_sortwordseq:NN`

```
\ExplSyntaxOn
```

Scratch variables.

```

\dim_new:N \l__glossaries_tmpa_dim
\dim_new:N \l__glossaries_tmpb_dim
\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
}
{
  \tl_if_exist:cTF
  { glo@ \use:c { glo@ #1 @parent } @ childcount }
  {

```

Increment child count.

```

\tl_set:ce
{ glo@ \use:c { glo@ #1 @parent } @ childcount }
{
  \int_eval:n
  {
    \c_one_int +
    \use:c { glo@ \use:c { glo@ #1 @parent } @ childcount }
  }
}
}
{

```

Append marker to ensure that child entries are placed immediately after parent entry.

```

\exp_args:NNv \glossaries_adjust_parent_sort:Nn
\l__glossaries_sort_value_tl
{ glo@ #1 @parent }
\tl_set:cn
{ glo@ \use:c { glo@ #1 @parent } @childcount }
{ 1 }
\prop_put:NvV \l__glossaries_sort_prop
{ glo@ #1 @parent }
\l__glossaries_sort_value_tl

```

```

    }
  }
}
{

```

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 }
    }

```

```

  }
}

```

Adjust parent sort value to help ensure hierarchy is preserved. (Arguments: token list variable containing sort value and parent label.) Need to ensure that parents with duplicate sort values aren't separated from their child entries.

```

\cs_new:Nn \glossaries_adjust_parent_sort:Nn
{
  \tl_put_right:Nn #1
  { \datatoolctrlboundary \datatoolasciistart #2 \datatoolasciistart }
}

```

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 }
  \bool_if:NT \l__glossaries_noidx_defer_sort_preprocess_bool
  {
    \exp_args:NNV \datatool_sort_preprocess:Nn
      #2 \l__glossaries_sort_value_tl
  }
  \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 }
  \bool_if:NT \l__glossaries_noidx_defer_sort_preprocess_bool
  {
    \exp_args:NNV \datatool_sort_preprocess:Nn
      #2 \l__glossaries_sort_value_tl
  }
  \exp_args:NV \DTLsortletterhandler
    \l__glossaries_sort_value_tl #2
}
```

Version 4.59: Case-sensitive word sort:

```
\cs_new:Nn \__glossaries_sort_wordcase:n
{
  \__glossaries_do_sort:Nn
    \__glossaries_sort_word_case_handler:nN
    { #1 }
}
\cs_new:Nn \__glossaries_sort_word_case_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \bool_if:NT \l__glossaries_noidx_defer_sort_preprocess_bool
  {
```



```

        \exp_args:NNV \datatool_sort_preprocess:Nn
        #2 \l__glossaries_sort_value_tl
    }
    \exp_args:NV \DTLsortwordcasehandler
    \l__glossaries_sort_value_tl #2
}

```

Version 4.59: Case-sensitive letter sort:

```

\cs_new:Nn \__glossaries_sort_lettercase:n
{
    \__glossaries_do_sort:Nn
    \__glossaries_sort_letter_case_handler:nN
    { #1 }
}
\cs_new:Nn \__glossaries_sort_letter_case_handler:nN
{
    \__glossaries_fetch_sort:n { #1 }
    \bool_if:NT \l__glossaries_noidx_defer_sort_preprocess_bool
    {
        \exp_args:NNV \datatool_sort_preprocess:Nn
        #2 \l__glossaries_sort_value_tl
    }
    \exp_args:NV \DTLsortlettercasehandler
    \l__glossaries_sort_value_tl #2
}

```

Case-sensitive non-localised (character code).

```

\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 }
    \bool_set_false:N \l_datatool_sort_to_lowercase_bool
    \bool_if:NT \l__glossaries_noidx_defer_sort_preprocess_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 }
  \bool_set_true:N \l_datatool_sort_to_lowercase_bool
  \bool_if:NT \l__glossaries_noidx_defer_sort_preprocess_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: $\langle handler\text{-}cs \rangle \{ \langle type \rangle \}$

```

\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:nn \__glossaries_actual_label:nn
      \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:nnn
  {
    \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_wordcase:n
{
  \PackageError {glossaries}
  {
    sort = wordcase ~ requires ~ datatool ~ v3.0+
  }
  {
    You ~ need ~ to ~ update ~ your ~ version ~ of ~ datatool ~
    to ~ use ~ 'wordcase' ~ sort ~ method
  }
}
\cs_new:Nn \__glossaries_sort_letter:n
{
  \@glo@sortentries{\@glo@sorthandler@letter}{#1}
}
\cs_new:Nn \__glossaries_sort_lettercase:n
{
  \PackageError {glossaries}
  {
    sort = wordcase ~ requires ~ datatool ~ v3.0+
  }
  {
    You ~ need ~ to ~ update ~ your ~ version ~ of ~ datatool ~

```

```

        to ~ use ~ 'wordcase' ~ sort ~ method
    }
}
\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' }
        {
            Check ~ the ~ spelling ~ of ~ the ~ sort=#1 ~ setting
        }
    }
}
\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.

```

\ifinlistcs{\@this@childlabel}{@glsref@#1}%
}%
{%
\listcsxadd{@glsref@#1}{\@this@childlabel}%
}%

```

Does this child have children?

```

\ifcsdef{@glo@sortingchildren@\@this@childlabel}%
{%
\@glo@addchildren{#1}{\@this@childlabel}%
}%
{%
}%
}%
\egroup
}

```

\@glo@do@sortentries

```

\newcommand*{\@glo@do@sortentries}[1]{%
\ifglshasparent{#1}%
{%

```

This entry has a parent, so add it to the child list

```

\protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
\ifcsundef{@glo@sortingchildren@\@glo@parent}%
{%
\csdef{@glo@sortingchildren@\@glo@parent}{}%
}%
}%

```

```

\expandafter\@glo@sortedinsert
\csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
Has the parent been added?
\ifinlistcs{\@glo@parent}{\@glsref@\@glo@type}%
{%
Yes, it has so do nothing.
}%
{%
No, it hasn't so add it now.
\expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
}%
}%
{%
\@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}

```

```
\@glo@sortedinsert{\<list>}{\<entry label>}
```

```
\@glo@sortedinsert
```

Insert into list.

```

\newcommand*{\@glo@sortedinsert}[2]{%
\dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%

```

The sort handlers need to be in the form required by datatool's `\dtl@sortlist` macro. These must set the count register `\dtl@sortresult` to either -1 ($\#1$ less than $\#2$), 0 ($\#1 = \#2$) or $+1$ ($\#1$ greater than $\#2$).

```
\@glo@sorthandler@word
```

```

\newcommand*{\@glo@sorthandler@word}[2]{%
\letcs\@gls@sort@A{glo\@glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{glo\@glsdetoklabel{#2}@sort}%
\edef\@glo@do@compare{%
\noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
{\expandonce\@gls@sort@B}%
{\expandonce\@gls@sort@A}%
}%
\@glo@do@compare
}

```

```
\@glo@sorthandler@letter
```

```

\newcommand*{\@glo@sorthandler@letter}[2]{%
\letcs\@gls@sort@A{glo\@glsdetoklabel{#1}@sort}%
\letcs\@gls@sort@B{glo\@glsdetoklabel{#2}@sort}%
\edef\@glo@do@compare{%
\noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
}

```

```

        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}

\@glo@sorthandler@case Case-sensitive sort.
\newcommand*{\@glo@sorthandler@case}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}

\@glo@sorthandler@nocase Case-insensitive sort.
\newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}

\@glo@sortmacro@word Sort macro for ‘word’
\newcommand*{\@glo@sortmacro@word}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@@sortentries{word}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=word]}{ }%
    }%
}

\@glo@sortmacro@wordcase Sort macro for ‘word’ (case-sensitive).
\newcommand*{\@glo@sortmacro@wordcase}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@@sortentries{wordcase}{#1}%
    }%
}

```

```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=wordcase]}{}}%
    }%
  }

```

\@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@lettercase Sort macro for ‘letter’ (case-sensitive).

```

\newcommand*{\@glo@sortmacro@lettercase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@@sortentries{lettercase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=lettercase]}{}}%
  }%
}

```

\@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{}%
  \for@gl@sentries[#1]{\@gl@thislabel}%
  {%
    \xifinlistcs{\@gl@thislabel}{\@gl@sref@#1}%
    {%
      \listadd{\@glo@sortinglist}{\@gl@thislabel}%
    }%
    {%
      Hasn’t been referenced.
    }%
  }%
  \cslet{\@gl@sref@#1}{\@glo@sortinglist}%
}

\@glo@sortmacro@def@do This won’t include parent entries that haven’t been referenced.

\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@gl@sref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@gl@sref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
}

```

```

    }%
    {}%
}

```

`\@glo@sortmacro@use` Sort macro for ‘use’. (No sorting is required, as the entries are already in order of use, so do nothing.)

```
\newcommand*{\@glo@sortmacro@use}[1]{}
```

`\@print@noidx@glossary` Glossary handler for `\printnoidx@glossary` which doesn’t use an indexing application. Since `\printnoidx@glossary` 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 `\printunsrt@glossary`) and check for `dtlsortgroup` 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 re-defining `\@print@noidx@glossary` to use `\@old@print@noidx@glossary` instead.

```

\ExplSyntaxOn
\newcommand*{\@print@noidx@glossary}{%
  \__glossaries_print_noidx:
}

```

New function:

```

\tl_new:N \l__glossaries_content_tl
\tl_new:N \l__glossaries_entry_prev_letter_group_tl
\cs_new:Nn \__glossaries_print_noidx:
{
  \tl_clear:N \l__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’ }
  {
    Check ~ the ~ spelling ~ of ~ the ~ sort={\@glo@sorttype} ~ setting
  }
}
\tl_clear:N \l__glossaries_entry_letter_group_tl
\tl_clear:N \l__glossaries_entry_prev_letter_group_tl

```

Version 4.59: initial hook:

```
\glsnoidxinithook
```

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 }
  \GlsNoIdxMissingAction { \@glo@type }
}
}
%\end{macro}
```

`\GlsNoIdxMissingAction` This needs to produce some output to allow for a document that only contains `\printnoidxglossary` (with `\glsaddall`).

```
\newcommand\GlsNoIdxMissingAction[1]{
  \glossaries_if_glossary_empty:nTF { #1 }
  { \strut }
  { \glossarysection [ \glossarytoctitle ] { \glossarytitle } }
}
```

Hooks:

`\glsnoidxinithook` Initial hook.

```
\newcommand \glsnoidxinithook { }
```

```
\glsnoidxitemhook{<level>}{<entry-label>}
```

\glsnoidxitemhook

Hook performed for each item added to the glossary token list.

```
\newcommand \glsnoidxitemhook [2] { }
```

\glsnoidxprecontenthook

```
\newcommand\glsnoidxprecontenthook{}
```

New handler macro for appending content:

```
\cs_new:Nn \__glossaries_noidx_do:n
{
  \__glossaries_noidx_add_to_listed:n { #1 }
  \int_set:Nn \gls@level { \use:c { glo@\glsdetoklabel{#1}@level } }
  \glsnoidxitemhook { \gls@level } { #1 }
  \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.

```
    \tl_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:N \@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
\tl_if_exist:cT
```

```

{ glo@ #1 @dtlsortgroup }
{
\tl_set:Ne \l__glossaries_entry_letter_group_tl
{
\exp_not:v { glo@ #1 @dtlsortgroup }
}
\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 \__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_group_title_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_group_title_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:N \@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{\@gls@ref@\@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.

```
\@gls@noref@warn{\@glo@type}%
\GlsNoIdxMissingAction { \@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`

`\glsnoidxloclist{<list cs>}`

Display location list.

```

\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}

```

`\glsnoidxloclisthandler` Handler for location list iterator.

```

\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}

```

Same as previous location so skip.

`\glsnoidxdisplayloclisthandler` Handler for location list iterator when used with `\glsdisplaynumberlist`.

```
\newcommand*\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
    Same as previous location so skip.
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
```

```
\glsnoidxdisplayloc{<prefix>}{<counter>}{<format>}{<location>}
```

`\glsnoidxdisplayloc`

Display a location in the location list.

```
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
```

```
\@gls@reference{<type>}{<label>}{<loc>}
```

`\@gls@reference`

Identifies that a reference has been used (for use in the aux file). All entries must be defined in the preamble.

```
\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 but first make sure that the internal list is defined.

```
\ifcsundef { glo@glsdetoklabel{#2} @loclist }
{ \csgdef { glo@glsdetoklabel{#2} @loclist } { } }
}%
```

Version 4.58: check if already in list. Note that `\ifinlist` can't be used for items with braces, which will be in the location.

```

\str_set:Nn \l__glossaries_current_location_str { #3 }
\cs_set_eq:NN \__glossaries_addloc:n \use:n
\forlistcsloop \__glossaries_check_loc_do:n
{ glo@ \glsdetoklabel{#2} @loclist }
\__glossaries_addloc:n
{ \listcsgadd { glo@ \glsdetoklabel{#2}@ loclist } { #3 } }
}

```

Version 4.58 new:

```

\str_new:N \l__glossaries_current_location_str
\cs_new:Nn \__glossaries_check_loc_do:n
{
\str_if_eq:VnT \l__glossaries_current_location_str { #1 }
{
\cs_set_eq:NN \__glossaries_addloc:n \use_none:n
\listbreak
}
}

```

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 }
}
}

```

Need to take into account that the entry may no longer be defined.

```

\bool_lazy_and:nnT
{ \tl_if_exist_p:c { glo@ #2 @parent } }
{ \bool_not_p:n { \tl_if_empty_p:c { glo@ #2 @parent } } }
{
\exp_args:Nv \__glossaries_noidx_add_parent_of_glsrefd:n
{ glo@ #2 @parent }
}
}
}

```

Keep track of parents of entries that were indexed in the previous run. (They may not have been indexed.)

```

\seq_new:N \g__glossaries_noidx_parents_of_glsrefd_seq
\cs_new:Nn \__glossaries_noidx_add_parent_of_glsrefd:n
{
\seq_if_in:NnF

```

```

\g__glossaries_noidx_parents_of_glsrefd_seq
{ #1 }
{
  \seq_gput_right:Nn
    \g__glossaries_noidx_parents_of_glsrefd_seq
    { #1 }

```

If entry has a parent, make sure that is also added.

```

\tl_if_empty:cF { glo@ #1 @parent }
{
  \exp_args:Nv \__glossaries_noidx_add_parent_of_glsrefd:n
    { glo@ #1 @parent }
}
}

```

Get number of entries that had references found in the aux file for a given glossary.

```

\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 }
}
\prg_new_conditional:Npnn \__glossaries_if_has_glsref:n #1
{ T, F , TF }
{
  \ifinlistcs { #1 } { @glsref@ \glsentrytype { #1 } }
  { \prg_return_true: }
  { \prg_return_false: }
}

```

```

\prg_new_conditional:Npnn \__glossaries_if_has_glsref_or_parent_of:n #1
{ T, F , TF }
{
  \seq_if_in:NnTF
    \g__glossaries_noidx_parents_of_glsrefd_seq
    { #1 }
    { \prg_return_true: }
    {
      \ifinlistcs { #1 } { @glsref@ \glsentrytype { #1 } }
      { \prg_return_true: }
      { \prg_return_false: }
    }
}
\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@dotocitle\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@dotocitle\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}{}%
\renewcommand*{\gls@currentglossaryxrlabel}{%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\renewcommand*{\gls@currentglossaryxrlabel}{%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{%
\label{\glsautoprefix\@glo@type}%
\glossary@write@seclabel{\glsautoprefix\@glo@type}%
\renewcommand*{\gls@currentglossaryxrlabel{\glsautoprefix\@glo@type}%
}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
```

```

\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}%
\@glossary@write@seclabel{\glsautoprefix\@glo@type}%
\renewcommand*\@gls@currentglossaryxrlabel{\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 \MessageBreak or \string\makenoidxglossaries\space
  is missing}%
  {The 'sort' key may only be used with
  \string\printnoidxglossary\space (which requires
  \string\makenoidxglossaries)}%
}
```

`\@@glo@assign@sortkey` For use with `\printnoidxglossary`

```
\newcommand*{\@@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
```

`\@glsnonetxpages` 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 `\glsnonetxpages` 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*{\@glsnonetxpages}{%
  \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}
```

`\glsnonetxpages` Outside of `\printglossary` this does nothing.

```
\newcommand*{\glsnonetxpages}{}%
```

`\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}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
```

`\glsstepentry` Advance the glossaryentry counter if in use. The argument is the label associated with the entry.

```
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
```

`\glsstepsubentry` Advance the glossarysubentry counter if in use. The argument is the label associated with the subentry.

```
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter

    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
```

`\glsrefentry` Reference the entry or sub-entry counter if in use, otherwise just do `\gls`.

```
\newcommand*{\glsrefentry}[1]{%
  \ifglsentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
}
```



```

\fi
}

```

`\glentrycounterlabel` Defines how to display the `glossaryentry` counter.

```

\newcommand*{\glentrycounterlabel}{%
\ifglentrycounter
\theglossaryentry.\space
\fi
}

```

`\glssubentrycounterlabel` Defines how to display the `glossarysubentry` counter.

```

\newcommand*{\glssubentrycounterlabel}{%
\ifglssubentrycounter
\theglossarysubentry)\space
\fi
}

```

`\glentryitem` Step and display `glossaryentry` counter, if appropriate.

```

\newcommand*{\glentryitem}[1]{%
\ifglentrycounter
\glstepentry{#1}\glentrycounterlabel
\else
\glresetsubentrycounter
\fi
}

```

`\glssubentryitem` Step and display `glossarysubentry` counter, if appropriate.

```

\newcommand*{\glssubentryitem}[1]{%
\ifglssubentrycounter
\glstepsubentry{#1}\glssubentrycounterlabel
\fi
}

```

`theglossary` (*env.*) If the `theglossary` environment has already been defined, a warning will be issued. This environment should be redefined by glossary styles.

```

\ifcsundef{theglossary}%
{%
\newenvironment{theglossary}{}{}%
}%
{%
\@gls@warnontheglossdefined
\renewenvironment{theglossary}{}{}%
}

```

The glossary header is given by `\glossaryheader`. This forms part of the glossary style, and must indicate what should appear immediately after the start of the `theglossary` environment. (For example, if the glossary uses a tabular-like environment, it may be used to set the header row.) Note that if you don't want a header row, the glossary style must redefine `\glossaryheader` to do nothing.

`\glossaryheader`

```
\newcommand*\glossaryheader{}
```

`\glstarget`

```
\glstarget{<label>}{<name>}
```

Provide user interface to `\@glstarget` to make it easier to modify the glossary style in the document.

```
\newcommand*\glstarget[2]{\@glstarget{\glolinkprefix#1}{#2}}
```

As from version 3.08, glossary information is now written to the external files using `\glossentry` and `\subglossentry` instead of `\glossaryentryfield` and `\glossarysubentryfield`. The default definition provides backward compatibility for glossary styles that use the old forms.

`\compatibleglossentry`

```
\compatibleglossentry{<label>}{<page-list>}
```

Removed in v4.50.

`\glossentry`

```
\glossentry{<label>}{<page-list>}
```

```
\newcommand*\glossentry[2]{\glossentryname{#1} #2}
```

`\glossentryname`

```
\newcommand*\glossentryname[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
```

`\Glossentryname`

```
\newcommand*\Glossentryname[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
```

`\GLOSSentryname`

```
\newcommand*\GLOSSentryname[1]{%
  \glsdoifexistsorwarn{#1}%
}
```

```

        {%
        \glsnamefont{\glsupercase{\glsentryname{#1}}}%
        }%
    }
    \glsmfuexcl{\GLOSSentryname}

\GlossEntryName
    \newcommand*{\GlossEntryName}[1]{%
    \glsnamefont{\glsentrytitlecase{#1}{name}}%
    }
    \glsmfuexcl{\GlossEntryName}

\glossentrydesc
    \newcommand*{\glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsentrydesc{#1}%
    }%
    }

\Glossentrydesc
    \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \Glsentrydesc{#1}%
    }%
    }
    \glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}

\GLOSSentrydesc
    \newcommand*{\GLOSSentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsupercase{\glsentrydesc{#1}}%
    }%
    }
    \glsmfuexcl{\GLOSSentrydesc}

\GlossEntryDesc NB don't use with multi-paragraph descriptions.
    \newcommand*{\GlossEntryDesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
        \glsentrytitlecase{#1}{desc}%
    }%
    }
    \glsmfuexcl{\GlossEntryDesc}

\glossentrysymbol
    \newcommand*{\glossentrysymbol}[1]{%

```

```

\glsdoifexistsorwarn{#1}%
{%
  \glsentrysymbol{#1}%
}%
}

```

`\Glossentrysymbol`

```

\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}

```

`\GLOSSentrysymbol`

```

\newcommand*{\GLOSSentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsuppercase{\glsentrysymbol{#1}}%
  }%
}
\glsmfuexcl{\GLOSSentrysymbol}

```

`\GlossEntrySymbol`

```

\newcommand*{\GlossEntrySymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrytitlecase{#1}{symbol}%
  }%
}
\glsmfuexcl{\GlossEntrySymbol}

```

`\compatiblesubglossentry{<level>}{<label>}{<page-list>}`

`\compatiblesubglossentry`

Removed in v4.50

`\setglossentrycompatibility`

`\glossaryentryfield{<label>}{<name>}{<description>}{<symbol>}{<page-list>}`

`\glossaryentryfield`

This command formerly governed how each entry row should be formatted in the glossary. Now deprecated. Removed in v4.50

`\glossarysubentryfield`

```
\glossarysubentryfield{<level>}{<label>}{<name>}{<description>}{<symbol>}{<page-list>}
```

This command governs how each subentry should be formatted in the glossary. Glossary styles need to redefine this command. Most of the predefined styles ignore *<symbol>*. The first argument is a number indicating the level. (The level should be greater than or equal to 1.)

Removed in v4.50.

`\subglossentry`

```
\subglossentry{<level>}{<label>}{<page-list>}
```

```
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
```

Within each glossary, the entries form distinct groups which are determined by the first character of the *sort* key. When using `makeindex`, there will be a maximum of 28 groups: symbols, numbers, and the 26 alphabetical groups A, ..., Z. If you use `xindy` the groups will depend on whatever alphabet is used. This is determined by the language or custom alphabets can be created in the `xindy` style file. The command `\glsgroupskip` specifies what to do between glossary groups. Glossary styles must redefine this command. (Note that `\glsgroupskip` only occurs between groups, not at the start or end of the glossary.)

`\glsgroupskip`

```
\newcommand*{\glsgroupskip}{}%
```

Each of the 28 glossary groups described above is preceded by a group heading. This is formatted by the command `\glsgroupheading` which takes one argument which is the *label* assigned to that group (not the title). The corresponding labels are: `glsymbols`, `glsnumbers`, A, ..., Z. Glossary styles must redefine this command. (In between groups, `\glsgroupheading` comes immediately after `\glsgroupskip`.)

`\glsgroupheading`

```
\newcommand*{\glsgroupheading}[1]{}%
```

It is possible to “trick” `makeindex` into treating entries as though they belong to the same group, even if the terms don’t start with the same letter, by modifying the *sort* key. For example, all entries belonging to one group could be defined so that the *sort* key starts with an *a*, while entries belonging to another group could be defined so that the *sort* key starts with a *b*, and so on. If you want each group to have a heading, you would then need to modify the translation control sequences `\glsgetgrouptitle` and `\glsgetgrouplabel` so that the label is translated into the required title (and vice-versa).

`\glsgetgrouptitle{<label>}`

This command produces the title for the glossary group whose label is given by `<label>`. By default, the group labelled `glssymbols` produces `\glssymbolsgroupname`, the group labelled `glsnumbers` produces `\glsnumbersgroupname` and all the other groups simply produce their label. As mentioned above, the group labels are: `glssymbols`, `glsnumbers`, `A`, ..., `Z`. If you want to redefine the group titles, you will need to redefine this command. Languages other than English may produce labels that are non-expandable, so we need to check for that otherwise it will create a “missing `\endcsname` inserted” error.

```
\glsgetgrouptitle
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
```

`\@gls@getgrouptitle` Gets the group title specified by the label (first argument) and stores in the second argument, which must be a control sequence.

```
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
```

Note that `\dtl@ifsingle` (provided by `datatool`) is being phased out.

```
\def#2{#1}%
\regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
{ \ifcsundef{#1groupname}{-}{\letcs#2{#1groupname}} }
}
```

`\@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
```

Provide a way of saving the group information. This is primarily intended for external processors to collect the information if required (where `glossary-hypernav` commands are not used). The user-level commands are `\glswriteglossarygroup` and `\glswriteglossarysubgroup` which are governed by the `saveglossarygroups` option. These commands need to be added to the `\glsgroupheading` and `\glssubgroupheading` glossary style commands to have an effect. For the base `glossaries` package, only the `tree*` style supports these commands. The other base styles will need `glossaries-extra-stylemods` to add them.

```
\@gls@writeglossarygroup Syntax: {\group-label}{\group-title}
\newcommand\@gls@writeglossarygroup[2]{%
  \@gls@writeglossarysubgroup{0}{#1}{#2}%
}
```

```
\@gls@writeglossarysubgroup Syntax: {\level}{\parent}{\group-label}{\group-title}
\ExplSyntaxOn
\newcommand\@gls@writeglossarysubgroup[4]{%
  \@gls@provide@glossarygroup
```

The group title needs expanding in case it's a language-sensitive command which may have been protected.

```
\glossaries_write_subgroup:eeeeee
{ \@gls@currentglossaryxrlabel }
{ \@gls@type }
{ \int_eval:n { #1 } }
{ #2 } { #3 } { #4 }
}
\cs_new:Nn \glossaries_write_subgroup:nnnnnn
{
  \protected@write \@auxout { }
```

```

    {
      \token_to_str:N \glsSavedGlossaryGroup
      { #1 } { #2 } { #3 } { #4 } { #5 } { #6 }
    }
  }
\cs_generate_variant:Nn \glossaries_write_subgroup:nnnnnn { eeeeeee }
\ExplSyntaxOff

```

`\@gls@provide@glossarygroup`

```

\newcommand*{\@gls@provide@glossarygroup}{%
  \protected@write\@auxout{}{\string\providecommand\string\glsSavedGlossaryGroup[6]{}}%
  \global\let\@gls@provide@glossarygroup\relax
}

```

`\glsSavedGlossaryGroup`

```

\newcommand\glsSavedGlossaryGroup[6]{}

```

The command `\setentrycounter` sets the entry’s associated counter (required by `\glshypernumber` etc.) `\glslink` and `\glsadd` encode the `\glossary` argument so that the relevant counter is set prior to the formatting command.

`\setentrycounter` The prefix is an optional argument because this command originally only had one argument. The optional argument was added in v3.0.

```

\newcommand*{\setentrycounter}[2][{}]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}

```

The current glossary style can be set using `\setglossarystyle{<style>}`.

`\setglossarystyle`

```

\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

Set the default style if it’s not already set.

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}

```


`\glossarystyle` Deprecated in v3.08a. Removed in v4.50.

`\newglossarystyle` New glossary styles can be defined using:

```
\newglossarystyle{<name>}{<definition>}
```

The `<definition>` argument should redefine `\theglossary`, `\glossaryheader`, `\glsgroupheading`, `\glossaryentryfield` and `\glsgroupskip` (see [subsection 2.19](#) for the definitions of predefined styles). Glossary styles should not redefine `\glossarypreamble` and `\glossarypostamble`, as the user should be able to switch between styles without affecting the pre- and postambles.

```
\newcommand{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
```

`\renewglossarystyle` Code for this macro supplied by Marco Daniel.

```
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
```

Glossary entries are encoded so that the second argument to `\glossaryentryfield` is always specified as `\glsnamefont{<name>}`. This allows the user to change the font used to display the name term without having to redefine `\glossaryentryfield`. The default uses the surrounding font, so in the list type styles (which place the name in the optional argument to `\item`) the name will appear in bold.

`\glsnamefont`

```
\newcommand*{\glsnamefont}[1]{#1}
```

Each glossary entry has an associated number list (usually page numbers) that indicate where in the document the entry has been used. The format for these number lists can be changed using the `format` key in commands like `\glslink`. The default format is given by `\glshypernumber`. This takes a single argument which may be a single number, a number range or a number list. The

number ranges are delimited with `\delimR`, the number lists are delimited with `\delimN`.

If the document doesn't have hyperlinks, the numbers can be displayed just as they are, but if the document supports hyperlinks, the numbers should link to the relevant location. This means extracting the individual numbers from the list or ranges. The `hyperref` package does this with the `\hyperpage` command, but this is encoded for comma and dash delimiters and only for the page counter, but this code needs to be more general. So I have adapted the code used in the `hyperref` package.

```
\glshypernumber
\ifcsundef{hyperlink}%
{%
\def\glshypernumber#1{#1}%
}%
{%
\def\glshypernumber#1{\@xp@glshypernumber{#1}}
}
```

```
\@xp@glshypernumber
\newrobustcmd{\@xp@glshypernumber}[1]{%
\@glshypernumber#1\nohyperpage{}}\@nil
}
```

`\@glshypernumber` This code was provided by Heiko Oberdiek to allow material to be attached to the location.

```
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
\ifx\#1\%
\else
\gls@delimR#1\delimR\delimR\%
\fi
\ifx\#2\%
\else
#2%
\fi
\ifx\#3\%
\else
\@glshypernumber#3\@nil
\fi
}
```

`\gls@delimR` displays a range of numbers for the counter whose name is given by `\@gls@counter` (which must be set prior to using `\glshypernumber`).

```
\gls@delimR
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
\gls@delimN{#1}%
\else
```

```

\@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}

```

`\gls@delimN` displays a list of individual numbers, instead of a range:

```

\gls@delimN
\def\gls@delimN#1{\gls@@delimN#1\delimN \delimN\}
\def\gls@@delimN#1\delimN #2\delimN#3\{\%
\ifx\#3\%
\@gls@numberlink{#1}%
\else
\@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}

```

The following code is modified from `hyperref`'s `\HyInd@pagelink` where the name of the counter being used is given by `\@gls@counter`.

```

\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup
}

\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%

\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else

\bgroup

```

Allow any markup to be stripped from the location when forming the target name.

```

\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glstrycounter\@glo@counterprefix
\glswrglosslocationtarget{\@gls@x}}%

```

If anything hasn't been fully expanded by this point then it will cause a problem, so sanitize to prevent an error, but it's likely the target won't be valid.

```

\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{\noexpand\glswrglosslocationtextfmt{\the\toks@}}}%
}%
\@gls@x
\fi
\else

```

```

\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}

```

The following commands will switch to the appropriate font, and create a hyperlink, if hyperlinks are supported. If hyperlinks are not supported, they will just display their argument in the appropriate font.

```

\hyperrm
\newcommand*{\hyperrm}[1]{\textrm{\glshypernumber{#1}}}

\hypersf
\newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}

\hypertt
\newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}

\hyperbf
\newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}

\hypermd
\newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}

\hyperit
\newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}

\hypersl
\newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}

\hyperup
\newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}

\hypersc
\newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}

\hyperemph
\newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}

```

2.17 Acronyms

`\oldacronym`

```
\oldacronym[⟨label⟩]{⟨abbrv⟩}{⟨long⟩}{⟨key-val list⟩}
```

This emulates the way the old `glossary` package defined acronyms. It is equivalent to `\newacronym[⟨key-val list⟩]{⟨label⟩}{⟨abbrv⟩}{⟨long⟩}` and it additionally defines the command `\⟨label⟩` which is equivalent to `\gls{⟨label⟩}` (thus

$\langle label \rangle$ 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 needed to cancel it out.

```
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
```

If `garamondx` has been loaded, need to use `\textulc` instead of `\textup`.

`\glstextup`

```
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
```

The following are defined for compatibility with version 2.07 and earlier.

`\glsshortkey`

```
\newcommand*{\glsshortkey}{short}
```

`\glsshortpluralkey`

```
\newcommand*{\glsshortpluralkey}{shortplural}
```

`\glslongkey`

```
\newcommand*{\glslongkey}{long}
```

`\glslongpluralkey`

```
\newcommand*{\glslongpluralkey}{longplural}
```

Using `\acrlinkfullformat` and `\acrfullformat` is now deprecated. However, these commands are still used in the initial definition of `\acrfull` and `\glsentryfull`. The acronym style now redefines `\glsentryfull`, and

`\acrfull` now uses `\acrfullfmt`, which is redefined by the acronym style. Similar for the case and plural variants. This means that `\acrlinkfullformat` and `\acrfullformat` should now longer be used with v4.50, which will ensure that `\setacronymstyle` is used on the first instance of `\newacronym` if a style hasn't been set.

Outlier cases may occur where someone has defined an entry explicitly with `\newglossaryentry` and set the long and short keys and is using `\glentryfull` or `\acrfull` without a style set. So, for now, the deprecated `\acrlinkfullformat` and `\acrfullformat` are retained even though the remaining obsolete acronym code has been removed. However, they may be removed in a future version.

`\acrlinkfullformat` Format for full links like `\acrfull`. Syntax: `\acrlinkfullformat{<long cs>}{<short cs>}{<options>}{<label>}{<insert>}` Deprecated.

```
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
```

`\acrfullformat` Default full form is *<long>* (*<short>*). Deprecated.

```
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
```

`\glsspace` Robust space to ensure it's written to the `.glsdefs` file.

```
\newrobustcmd{\glsspace}{\space}
```

The following `\acrfullfmt` set of commands are used by the corresponding `\acrfull` command, and are redefined by the acronym style to match the style's full form.

`\acrfullfmt` No case change full format. This default will be overridden by the acronym style, so this definition shouldn't actually be used any more, but it still needs to be defined so the it doesn't cause a problem when acronym styles redefine it. This is initialised to use the deprecated `\acrlinkfullformat` but will be redefined by the acronym style when it's set.

```
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
```

`\Acrfullfmt` Sentence case full format.

```
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
```

`\ACRfullfmt` All upper case full format.

```
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
```

```

\acrfullplfmt No case change plural full format.
    \newcommand*{\acrfullplfmt}[3]{%
        \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
    }

\Acrfullplfmt Sentence case plural full format.
    \newcommand*{\Acrfullplfmt}[3]{%
        \acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
    }

\ACRfullplfmt All upper case plural full format.
    \newcommand*{\ACRfullplfmt}[3]{%
        \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
    }

\acrfull Full form of the acronym.
    \newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}

    \newcommand*\ns@acrfull[2][{}]{%
        \new@ifnextchar[{\@acrfull{#1}{#2}}{%
            {\@acrfull{#1}{#2}}[]}%
    }

\@acrfull Low-level macro:
    \def\@acrfull#1#2[#3]{%
        Make it easier for acronym styles to change this:
        \acrfullfmt{#1}{#2}{#3}%
    }

    Default format for full acronym

\Acrfull
    \newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
    Add sentence case mapping.
    \glsmfuaddmap{\acrfull}{\Acrfull}

    \newcommand*\ns@Acrfull[2][{}]{%
        \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
            {\@Acrfull{#1}{#2}}[]}%
    }

    Low-level macro:
    \def\@Acrfull#1#2[#3]{%
        Make it easier for acronym styles to change this:
        \Acrfullfmt{#1}{#2}{#3}%
    }

\ACRfull
    \newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}

```



```

\newcommand*\ns@ACRfull[2][{}]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}

Low-level macro:
\def\@ACRfull#1#2[#3]{%
Make it easier for acronym styles to change this:
  \ACRfullfmt{#1}{#2}{#3}%
}

Plural:

\acrfullpl
\newrobustcmd*\acrfullpl{\@gls@hyp@opt\ns@acrfullpl}

\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}

Low-level macro:
\def\@acrfullpl#1#2[#3]{%
Make it easier for acronym styles to change this:
  \acrfullplfmt{#1}{#2}{#3}%
}

\Acrfullpl
\newrobustcmd*\Acrfullpl{\@gls@hyp@opt\ns@Acrfullpl}

Add sentence case mapping.
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}

\newcommand*\ns@Acrfullpl[2][{}]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[]}%
}

Low-level macro:
\def\@Acrfullpl#1#2[#3]{%
Make it easier for acronym styles to change this:
  \Acrfullplfmt{#1}{#2}{#3}%
}

\ACRfullpl
\newrobustcmd*\ACRfullpl{\@gls@hyp@opt\ns@ACRfullpl}

\newcommand*\ns@ACRfullpl[2][{}]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[]}%
}

```

Low-level macro:

```
\def\ACRfullpl#1#2[#3]{%
```

Make it easier for acronym styles to change this:

```
\ACRfullplfmt{#1}{#2}{#3}%  
}
```

2.18 Predefined acronym styles

`\acronymfont` Used by the acronym styles to apply a font change to the short form.

```
\newcommand{\acronymfont}[1]{#1}
```

`\firstacronymfont` Used by the acronym styles to apply a font change to the short form on first use.

```
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
```

`\acrnameformat` The styles that allow an additional description use `\acrnameformat{<short>}{<long>}` to determine what information is displayed in the name.

```
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
```

Define some tokens used by `\newacronym`:

`\glskeylisttok`

```
\newtoks\glskeylisttok
```

`\glslabeltok`

```
\newtoks\glslabeltok
```

`\glsshorttok`

```
\newtoks\glsshorttok
```

`\glslongtok`

```
\newtoks\glslongtok
```

`\newacronymhook` Provide a hook for `\newacronym`:

```
\newcommand*{\newacronymhook}{}
```

`\SetGenericNewAcronym` New improved version of setting the acronym style.

```
\newcommand*{\SetGenericNewAcronym}{%
```

Change the behaviour of `\Glsentryname` to workaround expansion issues that cause a problem for `\makefirstuc`

```
\let\@Gls@entryname\@Gls@acrentryname
```

Change the way acronyms are defined:

```
\renewcommand{\newacronym}[4][{}]{%  
  \ifdefempty{\@glsacronymlists}%  
  {%  
    \def\@glo@type{\acronymtype}%
```

```

\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
}%
{}%
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\protected@edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\expandonce{\acronymentry{##2}}},%
sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
text={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%

longplural={\the\glslongtok\noexpand\glspluralsuffix},%
\GenericAcronymFields,%
\the\glskeylisttok
}%
}%
\@do@newglossaryentry
}%

```

Make sure that \acrfull etc reflects the new style:

```

\renewcommand*{\acrfullfmt}[3]{%
\glslink{##1}{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink{##1}{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink{##1}{##2}{%
\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}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \Glsentrylong{##1}##2\space
    (\protect\firstacronymfont{\glsentryshort{##1}})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsentrylongpl{##1}##2\space
    (\protect\firstacronymfont{\glsentryshortpl{##1}})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \Glsentrylongpl{##1}##2\space

```

```

(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%

\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

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}{\glsacacfmt}{\glsacacentryfmt}%
}%
{%
\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}{\glsacrpluralsuffix}%
}

```

\glsacspace Space between long and short form for the above style. This uses a non-breakable space if the short form is less than 3em, otherwise it uses a regular space. It's unlikely that the short form will contain any instances of `\gls` so this just uses `\settowidth` instead of `\glsmeasurewidth`.

```

\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%

```

```

        \ifdim\dimen@<3em~\else\space\fi
    }

short-long <short> (<long>) acronym style.
    \newacronymstyle{short-long}%
    {%
Check for long form in case this is a mixed glossary.
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
    }%
    {%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
    \renewcommand*{\genacrfullformat}[2]{%
        \protect\firstacronymfont{\glsentryshort{##1}}##2\space
        (\glsentrylong{##1})%
    }%
    \renewcommand*{\Genacrfullformat}[2]{%
        \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
        (\glsentrylong{##1})%
    }%
    \renewcommand*{\genplacrfullformat}[2]{%
        \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
        (\glsentrylongpl{##1})%
    }%
    \renewcommand*{\Genplacrfullformat}[2]{%
        \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
        (\glsentrylongpl{##1})%
    }%

    \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
    \renewcommand*{\acronymsort}[2]{##1}%
    \renewcommand*{\acronymfont}[1]{##1}%
    \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%

    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
    }

long-sc-short <long> (\textsc{<short>}) acronym style.
    \newacronymstyle{long-sc-short}%
    {%
    \GlsUseAcrEntryDispStyle{long-short}%
    }%
    {%
    \GlsUseAcrStyleDefs{long-short}%
    \renewcommand{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
    }

long-sm-short <long> (\textsmaller{<short>}) acronym style.
    \newacronymstyle{long-sm-short}%

```

```

{
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}

sc-short-long <short> (\textsc{<long>}) acronym style.
\newacronymstyle{sc-short-long}%
{
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssupacrpluralsuffix}%
}

sm-short-long <short> (\textsmaller{<long>}) acronym style.
\newacronymstyle{sm-short-long}%
{
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}

long-short-desc <long> ({<short>}) acronym style that has an accompanying description (which
the user needs to supply).
\newacronymstyle{long-short-desc}%
{
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glssentrylong{##1}\space (\acronymfont{\glssentryshort{##1}})}%
  }

long-sp-short-desc <long> ({<short>}) acronym style that has an accompanying description (which
the user needs to supply). The space between the long and short form is given
by \glsspace.
\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}%
  {%
    \GlsUseAcrEntryDispStyle{long-sc-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sc-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
    }

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
  \newacronymstyle{long-sm-short-desc}%
  {%
    \GlsUseAcrEntryDispStyle{long-sm-short}%
  }%
  {%
    \GlsUseAcrStyleDefs{long-sm-short}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \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}%
  {%
    \GlsUseAcrEntryDispStyle{short-long}%
  }%
  {%
    \GlsUseAcrStyleDefs{short-long}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
  }

```

```

\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}

```

sc-short-long-desc *<long>* (`\textsc{<short>}`) acronym style that has an accompanying description (which the user needs to supply).

```

\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
{%
  \GlsUseAcrEntryDisplayStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
  }

```

dua *<long>* only acronym style.

```

\newacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
  \ifglshaslong{\glslabel}%
  {%
    \glsifplural
    {%

```

Plural form:

```

\glscapscase
{%

```

Plural form, don't adjust case:

```

\glsentrylongpl{\glslabel}\glsinsert

```

```

    }%
    {%

Plural form, make first letter upper case:
    \Glsentrylongpl{\glslabel}\glsinsert
    }%
    {%

Plural form, all caps:
    \glssupercase
    {\glsentrylongpl{\glslabel}\glsinsert}%
    }%
    }%
    {%

Singular form
    \glscapscase
    {%

Singular form, don't adjust case:
    \glsentrylong{\glslabel}\glsinsert
    }%
    {%

Subsequent singular form, make first letter upper case:
    \Glsentrylong{\glslabel}\glsinsert
    }%
    {%

Subsequent singular form, all caps:
    \glssupercase
    {\glsentrylong{\glslabel}\glsinsert}%
    }%
    }%
    }%
    {%

Not an acronym:
    \glsgenentryfmt
    }%
    }%
    {\glscustomtext\glsinsert}%
    }%
    {%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

    \renewcommand*{\acrfullfmt}[3]{%
        \glslink[##1]{##2}{\glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
    \renewcommand*{\Acrfullfmt}[3]{%
        \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glsentryshort{##2}})}%
    \renewcommand*{\ACRfullfmt}[3]{%

```

```

\glslink{##1}{##2}{%
  \glsuppercase{\glsentrylong{##2}##3\space
    (\acronymfont{\glsentryshort{##2}})}}}%

\renewcommand*{\acrfullplfmt}[3]{%
  \glslink{##1}{##2}{\glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}}%

\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink{##1}{##2}{\Glsentrylongpl{##2}##3\space
    (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink{##1}{##2}{%
    \glsuppercase{\glsentrylongpl{##2}##3\space
      (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}

```

`dua-desc` *<long>* only acronym style with user-supplied description.

```

\newacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%

  \renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%

```

`footnote` *<short>*\footnote{*<long>*} acronym style.

```

\newacronymstyle{footnote}%
{%

```

```

\ifglsbaselength{\glslabel}{\glsbaselength}{\glsbaselength}{\glsbaselength}%
}%
}%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

```

```

\glsyhyperfirstfalse
\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}{%
    \glssupercase{\acronymfont{\glsentryshort{##2}}##3\space
      (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink{##1}{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*{\AcRfullplfmt}[3]{%
  \glslink{##1}{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink{##1}{##2}{%
    \glssupercase{\acronymfont{\glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}}}%

```

Similarly for \glentryfull etc:

```
\renewcommand*\glentryfull}[1]{%
  \acronymfont{\glentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glentrylong{##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}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
{%
  \GlsUseAcrEntryDisplayStyle{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}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*\GenericAcronymFields{}%
  \renewcommand*\acronymsort[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]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##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]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
```

\DefineAcronymSynonyms

```
\newcommand*{\DefineAcronymSynonyms}{%
```

Short form

```
\acs
```

```
\let\acs\acrshort
```

First letter uppercase short form

```
\Acs
```

```
\let\Acs\Acrshort
```

Plural short form

```
\acsp
```

```
\let\acsp\acrshortpl
```

First letter uppercase plural short form

```
\Acsp
```

```
\let\Acsp\Acrshortpl
```

Long form

```
\acl
    \let\acl\acrlong
```

Plural long form

```
\aclp
    \let\aclp\acrlongpl
```

First letter upper case long form

```
\Acl
    \let\Acl\Acrlong
```

First letter upper case plural long form

```
\Aclp
    \let\Aclp\Acrlongpl
```

Full form

```
\acf
    \let\acf\acrfull
```

Plural full form

```
\acfp
    \let\acfp\acrfullpl
```

First letter upper case full form

```
\Acf
    \let\Acf\Acrfull
```

First letter upper case plural full form

```
\Acfp
    \let\Acfp\Acrfullpl
```

Standard form

```
\ac
    \let\ac\gls
```

First upper case standard form

```
\Ac
    \let\Ac\Gls
```

Standard plural form

```
\acp
    \let\acp\glspl
```


Standard first letter upper case plural form

`\Acp`

`\let\Acp\Glspl`

`}`

Define synonyms if required

`\ifglacrshortcuts`

`\DefineAcronymSynonyms`

`\fi`

`\SetDefaultAcronymDisplayStyle` Sets the default acronym display style for given glossary. Used by glossaries-extra when restoring `\newacronym`.

`\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%`

`\defglentryfmt[#1]{\glsgenentryfmt}%`

`}`

The following commands for setting the acronym style are now deprecated.

`\DefaultNewAcronymDef` Sets up the acronym definition for the default style. The information is provided by the tokens `\glslabeltok`, `\glsshorttok`, `\glslongtok` and `\glskeylisttok`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\expand@assign@default@plurals` `\@@gls@noexpand@assign@default@plurals` was used in `\DefaultNewAcronymDef`. Added in v4.47 to fix bug #162 (<https://www.dickimaw-books.com/bugtracker.php?key=162>). Removed in v4.50 along with the deprecated styles.

`\@@gls@@default@acr@plural`

`\gls@@default@acr@firstplural`

`\gls@@default@acr@descplural`

`\SetDefaultAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\acrfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

`\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}`

`\acrlinkfootnote` This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained.

`\newcommand*{\acrlinkfootnote}[3]{%`

`\footnote{\glslink[#1]{#2}{#3}}%`

`}`

<code>\acrnlkfootnote</code>	This was used by the old footnote acronym styles. Since it may have been used outside of those styles, it has been retained. <pre>\newcommand*{\acrnlkfootnote}[3]{% \footnote{#3}% }</pre>
<code>\FootnoteAcronymDisplayStyle</code>	Sets the acronym display style for given glossary for the description and footnote combination. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionFootnoteNewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionFootnoteAcronymStyle</code>	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 <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionDUAAcronymDisplayStyle</code>	Sets the acronym display style for given glossary with description and dua combination. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionDUANewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\SetDescriptionDUAAcronymStyle</code>	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 <code>\newacronymstyle</code> . Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionAcronymDisplayStyle</code>	Sets the acronym display style for given glossary using the description setting (but not footnote or dua). Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\DescriptionNewAcronymDef</code>	Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\SetDescriptionAcronymStyle</code>	Option description is used, but not dua or footnote. Store long form in first key and short form in text and symbol key. The name is stored using <code>\acrnameformat</code> to allow the user to override the way the name is displayed in the list of acronyms. Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.
<code>\FootnoteAcronymDisplayStyle</code>	Sets the acronym display style for given glossary with footnote setting (but not description or dua). Deprecated following the introduction of <code>\setacronymstyle</code> . Removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetFootnoteAcronymStyle` If footnote package option is specified, set the first use to append the long form (stored in `description`) as a footnote. Use the `description` key to store the long form. This style was deprecated with the introduction of `\newacronymstyle`.
 Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\glsdoparenifnotempty` Do a space followed by the argument if the argument doesn't expand to empty or `\relax`. If argument isn't empty (or `\relax`), apply the macro to it given in the second argument.

```

\newrobustcmd*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}

```

`\SetSmallAcronymDisplayStyle` Sets the acronym display style for given glossary where neither footnote nor description is required, but smallcaps or smaller specified. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SmallNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetSmallAcronymStyle` Neither footnote nor description required, but smallcaps or smaller specified. Use the `symbol` key to store the short form and `first` to store the long form. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetDUADisplayStyle` Sets the acronym display style for given glossary with dua setting. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\DUANewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetDUASStyle` Always expand acronyms. This style was deprecated with the introduction of `\newacronymstyle`. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\@gls@deprecated@acr@warning`

```

\newcommand*{\@gls@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.

```

```

    Use \string\setacronymstyle{#2} with just glossaries.sty or
    use \string\setabbreviationstyle[acronym]{#3} with
    glossaries-extra.sty}%
}

```

`\SetAcronymStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

Allow user to define their own custom acronyms. (For compatibility with versions before v3.0, the short form is stored in the user1 key, the plural short form is stored in the user2 key, the long form is stored in the user3 key and the plural long form is stored in the user4 key.) Defaults to displaying only the acronym with the long form as the description.

`\SetCustomDisplayStyle` Sets the acronym display style. Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\CustomAcronymFields` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\CustomNewAcronymDef` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

`\SetCustomStyle` Deprecated following the introduction of `\setacronymstyle`. Removed in v4.50.

2.19 Predefined Glossary Styles

The `glossaries` bundle comes with some predefined glossary styles. These need to be loaded now for the `style` option to use them.

First, the glossary hyper-navigation commands need to be loaded.

```
\RequirePackage{glossary-hypernav}
```

The styles that use list-like environments. These are not loaded if the `nolist` option is used:

```
\@gls@loadlist
```

The styles that use the `longtable` environment. These are not loaded if the `nolong` package option is used.

```
\@gls@loadlong
```

The styles that use the `supertabular` environment. These are not loaded if the `nosuper` package option is used or if the `supertabular` package isn't installed.

```
\@gls@loadsuper
```

The tree-like styles. These are not loaded if the `notree` package option is used.

```
\@gls@loadtree
```

The default glossary style is set according to the `style` package option, but can be overridden by `\glossarystyle`. The required style must be defined at this point.

```
\ifx\@glossary@default@style\relax
\else
\setglossarystyle{\@glossary@default@style}
\fi
```

2.20 Debugging Commands

`\showgloparent`

```
\showgloparent{\label}
```

```
\newcommand*\showgloparent[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
```

`\showglolevel`

```
\showglolevel{\label}
```

```
\newcommand*\showglolevel[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
```

`\showglotext`

```
\showglotext{\label}
```

```
\newcommand*\showglotext[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
```

`\showgloplural`

```
\showgloplural{\label}
```

```
\newcommand*\showgloplural[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
```

`\showglofirst`

```
\showglofirst{\label}
```

```
\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
```

\showglofirstpl

```
\showglofirstpl{\label{}}
```

```
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
```

\showgloftype

```
\showgloftype{\label{}}
```

```
\newcommand*{\showgloftype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
```

\showglocounter

```
\showglocounter{\label{}}
```

```
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
```

\showglouserii

```
\showglouserii{\label{}}
```

```
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
```

\showglouseriii

```
\showglouseriii{\label{}}
```

```
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
```

\showglouseriii

\showglouseriii{<label>}

```
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriii\endcsname
}
```

\showglouseriv

\showglouseriv{<label>}

```
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@useriv\endcsname
}
```

\showglouserv

\showglouserv{<label>}

```
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@userv\endcsname
}
```

\showglouservi

\showglouservi{<label>}

```
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@uservi\endcsname
}
```

\showgloname

\showgloname{<label>}

```
\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@name\endcsname
}
```

\showglodesc

\showglodesc{<label>}

```
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@desc\endcsname
}
```

\showglodescplural

\showglodescplural{<label>}

```
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@descplural\endcsname
}
```

\showglosort

\showglosort{<label>}

NB this may not be the same as the sortvalue internal field.

```
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@sort\endcsname
}
```

\showglosymbol

\showglosymbol{<label>}

```
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@symbol\endcsname
}
```

\showglosymbolplural

\showglosymbolplural{<label>}

```
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@symbolplural\endcsname
}
```

\showgloshort

\showgloshort{<label>}

```
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@short\endcsname
}
```

\showgloshortplural

\showgloshortplural{<label>}

```
\newcommand*{\showgloshortplural}[1]{%
  \expandafter\show\csname glo@glsdetoklabel{#1}@shortpl\endcsname
}
```


\showglolong

```
\showglolong{<label>}
```

```
\newcommand*{\showglolong}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname  
}
```

\showglolongplural

```
\showglolongplural{<label>}
```

```
\newcommand*{\showglolongplural}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longpl\endcsname  
}
```

\showgloindex

```
\showgloindex{<label>}
```

```
\newcommand*{\showgloindex}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname  
}
```

\showgloflag

```
\showgloflag{<label>}
```

```
\newcommand*{\showgloflag}[1]{%  
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname  
}
```

\showgloloclist

```
\showgloloclist{<label>}
```

```
\newcommand*{\showgloloclist}[1]{%  
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname  
}
```

\showglofield

```
\showglofield{<label>}{<field>}
```

```
\newcommand*{\showglofield}[2]{%  
  \csshow{glo@\glsdetoklabel{#1}@#2}%  
}
```

<code>\showacronymlists</code>	<div><code>\showacronymlists</code></div> <p>Show list of glossaries that have been flagged as a list of acronyms.</p> <pre>\newcommand*{\showacronymlists}{% \show\@glsacronymlists }</pre>
<code>\showglossaries</code>	<div><code>\showglossaries</code></div> <p>Show list of defined glossaries.</p> <pre>\newcommand*{\showglossaries}{% \show\@glo@types }</pre>
<code>\showglossaryin</code>	<div><code>\showglossaryin{<glossary-label>}</code></div> <p>Show the ‘in’ extension for the given glossary.</p> <pre>\newcommand*{\showglossaryin}[1]{% \expandafter\show\csname @glotype@#1@in\endcsname }</pre>
<code>\showglossaryout</code>	<div><code>\showglossaryout{<glossary-label>}</code></div> <p>Show the ‘out’ extension for the given glossary.</p> <pre>\newcommand*{\showglossaryout}[1]{% \expandafter\show\csname @glotype@#1@out\endcsname }</pre>
<code>\showglossarytitle</code>	<div><code>\showglossarytitle{<glossary-label>}</code></div> <p>Show the title for the given glossary.</p> <pre>\newcommand*{\showglossarytitle}[1]{% \expandafter\show\csname @glotype@#1@title\endcsname }</pre>
<code>\showglossarycounter</code>	<div><code>\showglossarycounter{<glossary-label>}</code></div> <p>Show the counter for the given glossary.</p> <pre>\newcommand*{\showglossarycounter}[1]{% \expandafter\show\csname @glotype@#1@counter\endcsname }</pre>

`\showglossaryentries`

```
\showglossaryentries{<glossary-label>}
```

Show the list of entry labels for the given glossary.

```
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glo@list@#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{v5.1}{2026-04-21}
```

Package Definition:

```
\ProvidesPackage{glossary-hypernav}[2026/04/21 v5.1 (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 $\langle text \rangle$ a hyperlink to the glossary group whose label is given by $\langle label \rangle$ for the glossary given by $\langle type \rangle$.

`\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 `\@gls@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@hypergroupelist@ $\langle glossary \rangle$` (where $\langle glossary \rangle$ is the glossary label). They are now stored in a sequence `\g_glossaries_ $\langle glossary \rangle$ _hypergroups_seq`.

```
\ExplSyntaxOn
```

```
\glossaries_add_hypergroup:nn{ $\langle glossary \rangle$ }{ $\langle target \rangle$ }
```

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 }
        }
    }
}
\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, `\@glo@type` is used which is set by `\printglossary` to the current glossary label. The optional argument complicates things for glossaries-extra.

```
\newcommand*{\glsnavhypertarget}{\protect\@gls@navhypertarget}
```

`\@gls@navhypertarget`

```
\newcommand*{\@gls@navhypertarget}[3][\currentglossary]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}
```

`\glsnavhypergroupdotarget` Provided to make it easier for glossaries-extra to adjust the target.

```
\newcommand{\glsnavhypergroupdotarget}[3]{
  \@glstarget{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
```

The actual code is now in an internal command that doesn't have an optional argument, which makes it easier to save and restore the original behaviour.

```

\@glsnavhypertarget
    \newcommand*{\@glsnavhypertarget}[3]{%
Add this group to the aux file for re-run check.
        \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
Add the target.
        \glsnavhypergroupdotarget { #1 } { #2 } { #3 }
Check list of known groups to determine if a re-run is required.
        \glossaries_if_hypergroup:nxF { #1 } { #2 }
        {
This group was not included in the list, so issue a warning.
            \GlossariesWarningNoLine{Navigation ~ panel ~
                for ~ glossary ~ type ~ ‘#1’^^Jmissing ~ group ~ ‘#2’}%
            \gdef\gls@hypergrouprerun{%
                \GlossariesWarningNoLine{Navigation ~ panel ~
                    has ~ changed. ~ Rerun ~ LaTeX}}%
        }
    }

\gls@hypergrouprerun Give a warning at the end if re-run required
    \let\gls@hypergrouprerun\relax
    \AtEndDocument{\gls@hypergrouprerun}

\@gls@hypergroup This adds to (or creates) the command \@gls@hypergrouplist@(glossary type)
which lists all groups for a given glossary, so that the navigation bar only con-
tains 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{v5.1}{2026-04-21}

```

Declare package:

```
\ProvidesPackage{glossary-inline}[2026/04/21 v5.1 (NLCT)]
```

`inline` Define the inline style.

```
\newglossarystyle{inline}{%
```

Start of glossary sets up first empty separator between entries. (This is then changed by `\glossentry`)

```

\renewenvironment{theglossary}{%
{%
\def\gls@inlinesep}{%

```

```

\def\gls@inlinesubsep{}%
\def\gls@inlinepostchild{}%
}%
{\glspostinline}%

```

No header:

```
\renewcommand*\glossaryheader{}%
```

No group headings (if heading is required, add `\glsinlinedopostchild` to start definition in case heading follows a child entry):

```
\renewcommand*\glsgroupheading}[1]{}%
```

Just display separator followed by name and description:

```

\renewcommand{\glossentry}[2]{%
  \glsinlinedopostchild
  \gls@inlinesep
  \glsentryitem{##1}%
  \glsinlinenameformat{##1}{%
    \glossentryname{##1}%
  }%
  \ifglsdescsuppressed{##1}%
  {%
    \glsinlineemptydescformat
    {%
      \glossentrysymbol{##1}%
    }%
    {%
      ##2%
    }%
  }%
}%
{%
  \ifglsdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\glsinlineifhaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
}%
\def\gls@inlinesep{\glsinlineseparator}%
}%

```

Sub-entries display description:

```

\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
}

```



```

\glssubentryitem{##2}%
\glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
\def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
Nothing special between groups:
\renewcommand*{\glsgroupskip}{}%
}

\glsinlineifhaschildren
\newcommand{\glsinlineifhaschildren}[3]{%
\ifgls@haschildren{#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{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-list}[2026/04/21 v5.1 (NLCT)]
```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
```

`\glslistgroupheaderfmt` Provide a way of adjusting the format of the group headings.

```
\newcommand*{\glslistgroupheaderfmt}[1]{#1}
```

`\glslistnavigationitem` Provide a way of adjusting the format of the navigation header. This puts the navigation line inside the optional argument of `item` to prevent unwanted space occurring at the start, but this can cause a problem if the navigation line is too long. With this command, it makes it easier for the user to customise the style without having to remember to modify `\glossaryheader` after the style has been set.

```
\newcommand*{\glslistnavigationitem}[1]{\item[#1]}
```

`\glslistinit` Provide a way to ensure appropriate expansion of the name if `entrycounter` is used with `getttitlestring`:

```
\newcommand{\glslistinit}{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
  {}%
}
```

`\glslistexpandedname`

```
\newcommand{\glslistexpandedname}[1]{%
\glsunexpandedfieldvalue{#1}{name}%
}
```

list The list glossary style uses the `description` environment. The group separator `\glsgroupskip` is redefined as `\indexspace` which produces a gap between groups. The glossary heading and the group headings do nothing. Sub-entries immediately follow the main entry without the sub-entry name. This style does not use the entry's symbol. This is used as the default style for the glossaries package.

```
\newglossarystyle{list}{%
```

Use `description` environment:

```
\renewenvironment{theglossary}%
{\glslistinit\begin{description}}{\end{description}}%
```

No header at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries start a new item in the list:

```
\renewcommand*{\glossentry}[2]{%
\item[\glsentryitem{##1}%
\glstarget{##1}{\glossentryname{##1}}]
\glossentrydesc{##1}\glspostdescription\space ##2}%
```

Sub-entries continue on the same line:

```
\renewcommand*{\subglossentry}[3]{%
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\space
\glossentrydesc{##2}\glspostdescription\space ##3.}%
```

Add vertical space between groups:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

listgroup The listgroup style is like the list style, but the glossary groups have headings.

```
\newglossarystyle{listgroup}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}%
```

listhypergroup The listhypergroup style is like the listgroup style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{listhypergroup}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%  
  \glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a hypertarget:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt  
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%  
}%  
}
```

altlist The altlist glossary style is like the list style, but places the description on a new line. Sub-entries follow in separate paragraphs without the sub-entry name. This style does not use the entry's symbol.

```
\newglossarystyle{altlist}{%
```

Base it on the list style:

```
\setglossarystyle{list}%
```

Main (level 0) entries start a new item in the list with a line break after the entry name:

```
\renewcommand*{\glossentry}[2]{%  
  \item[\glsentryitem{##1}%  
    \glstarget{##1}{\glossentryname{##1}}}%
```

Version 3.04 changed `\newline` to the following paragraph break stuff (thanks to Daniel Gebhardt for supplying the fix) to prevent a page break occurring at this point.

```
\mbox{} \par \nobreak \@afterheading  
\glossentrydesc{##1} \glspostdescription \space ##2}%
```

Sub-entries start a new paragraph:

```
\renewcommand{\subglossentry}[3]{%  
  \par  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut} \glossentrydesc{##2} \glspostdescription \space ##3}%  
}
```

altlistgroup The altlistgroup glossary style is like the altlist style, but the glossary groups have headings.

```
\newglossarystyle{altlistgroup}{%
```

Base it on the altlist style:

```
\setglossarystyle{altlist}%
```

Each group has a heading:

```
\renewcommand*{\glsgroupheading}[1]{%  
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}}
```

`altlisthypergroup` The `altlisthypergroup` glossary style is like the `altlistgroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{altlisthypergroup}{%
```

Base it on the `altlist` style:

```
\setglossarystyle{altlist}%
```

Add navigation links at the start of the environment.

```
\renewcommand*{\glossaryheader}{%
\glslistnavigationitem{\glsnavigation}}%
```

Each group has a heading with a `hypertarget`:

```
\renewcommand*{\glsgroupheading}[1]{%
\item[\glslistgroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]}
```

`listdotted` The `listdotted` glossary style was supplied by Axel Menzel. I've modified it slightly so that the distance from the start of the name to the end of the dotted line is specified by `\glslistdottedwidth`. Note that this style ignores the page numbers as well as the symbol. Sub-entries are displayed in the same way as top-level entries.

```
\newglossarystyle{listdotted}{%
```

Base it on the `list` style:

```
\setglossarystyle{list}%
```

Each main (level 0) entry starts a new item:

```
\renewcommand*{\glossentry}[2]{%
\item[\makebox[\glslistdottedwidth][l]{%
\glsentryitem{##1}%
\glstarget{##1}{\glossentryname{##1}}%
\unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
```

Sub entries have the same format as main entries:

```
\renewcommand*{\subglossentry}[3]{%
\item[\makebox[\glslistdottedwidth][l]{%
\glssubentryitem{##2}%
\glstarget{##2}{\glossentryname{##2}}%
\unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
}
```

```
\glslistdottedwidth
```

```
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
```

`sublistdotted` This style is similar to the `glostylelistdotted` style, except that the main entries just have the name displayed.

```
\newglossarystyle{sublistdotted}{%
```

Base it on the `listdotted` style:

```
\setglossarystyle{listdotted}%
```

Main (level 0) entries just display the name:

```
\renewcommand*{\glossentry}[2]{%
  \item[\glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]{%
}
```

3.4 Glossary Styles using longtable (glossary-long.sty)

The glossary styles defined in the glossary-long package used the longtable environment in the glossary.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-long-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-long-2021-11-01.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-long}[2026/04/21 v5.1 (NLCT)]
```

Requires the longtable package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. (There's a chance that the user may specify `nolong` and then load `glossary-long` later, in which case `\glsdescwidth` may have already been defined by `glossary-super`. The same goes for `\glspagelistwidth`.)

```
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column.

```
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{}
```

`long` The long glossary style command which uses the longtable environment:

```
\newglossarystyle{long}{%
```

Use longtable with two columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{\glsdescwidth}]}%
  {\end{longtable}}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupeheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside
`\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

longborder The `longborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longborder}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|p{\glstdescwidth}|}{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

longheader The `longheader` style is like the `long` style but with a header:

```
\newglossarystyle{longheader}{%
```

Base it on the `glostylelong` style:

```
\setglossarystyle{long}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
```

longheaderborder The `longheaderborder` style is like the `long` style but with a header and border:

```
\newglossarystyle{longheaderborder}{%
```

Base it on the `glostylelongborder` style:

```
\setglossarystyle{longborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries \entryname & \bfseries
  \descriptionname\tabularnewline\hline
  \endhead
  \hline\endfoot}%
}
```

`long3col` The `long3col` style is like `long` but with 3 columns

```
\newglossarystyle{long3col}{%
```

Use a `longtable` with 3 columns:

```
\renewenvironment{theglossary}%
  {\begin{longtable}[lp{\glstdescwidth}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} & ##2\tabularnewline
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```


long3colborder The long3colborder style is like the long3col style but with a border:

```
\newglossarystyle{long3colborder}{%
```

Base it on the glosstylelong3col 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 glosstylelong3col style:

```
\setglossarystyle{long3col}{%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

long3colheaderborder The long3colheaderborder style is like the above but with a border

```
\newglossarystyle{long3colheaderborder}{%
```

Base it on the glosstylelong3colborder style:

```
\setglossarystyle{long3colborder}{%
```

Set the table's header and add horizontal line at table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

long4col The long4col style has four columns where the third column contains the value of the associated symbol key.

```
\newglossarystyle{long4col}{%
```

Use a longtable with 4 columns:

```
\renewenvironment{theglossary}{%
  {\begin{longtable}{llll}}%
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside \glsgroupskip (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

long4colheader The long4colheader style is like long4col but with a header row.

```
\newglossarystyle{long4colheader}{%
```

Base it on the glostylelong4col style:

```
\setglossarystyle{long4col}%
```

Table has a header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

long4colborder The long4colborder style is like long4col but with a border.

```
\newglossarystyle{long4colborder}{%
```

Base it on the glostylelong4col style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%
{\begin{longtable}{|l|l|l|l|}}%
{\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`long4colheaderborder` The `long4colheaderborder` style is like the above but with a border.

```
\newglossarystyle{long4colheaderborder}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns surrounded by vertical lines:

```
\renewenvironment{theglossary}%
{\begin{longtable}{|l|l|l|l|}}%
{\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%
\hline\bfseries\entryname&\bfseries\descriptionname&
\bfseries \symbolname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
```

`altlong4col` The `altlong4col` style is like the `long4col` style but can have multiline descriptions and page lists.

```
\newglossarystyle{altlong4col}{%
```

Base it on the `glostylelong4col` style:

```
\setglossarystyle{long4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
{\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
{\end{longtable}}%
}
```

`altlong4colheader` The `altlong4colheader` style is like `altlong4col` but with a header row.

```
\newglossarystyle{altlong4colheader}{%
```

Base it on the `glostylelong4colheader` style:

```
\setglossarystyle{long4colheader}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
{\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
{\end{longtable}}%
}
```

`altlong4colborder` The `altlong4colborder` style is like `altlong4col` but with a border.

```
\newglossarystyle{altlong4colborder}{%
```

Base it on the `glostylelong4colborder` style:

```
\setglossarystyle{long4colborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|l|p{\glsgdescwidth}|l|p{\glspagelistwidth}|}%  
  {\end{longtable}}%  
}
```

`altlong4colheaderborder` The `altlong4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlong4colheaderborder}{%
```

Base it on the `glostylelong4colheaderborder` style:

```
\setglossarystyle{long4colheaderborder}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{|l|p{\glsgdescwidth}|l|p{\glspagelistwidth}|}%  
  {\end{longtable}}%  
}
```

3.5 Glossary Styles using `longtable` and `booktabs` (`glossary-longbooktabs.sty`)

The styles here are based on David Carlisle's patch at <http://tex.stackexchange.com/a/56890>

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-longbooktabs-2020-03-19.sty}  
\DeclareRelease{v4.49}{2021-11-01}{glossary-longbooktabs-2021-11-01.sty}  
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-longbooktabs}[2026/04/21 v5.1 (NLCT)]
```

Requires `booktabs` package:

```
\RequirePackage{booktabs}
```

and the base packages for long styles:

```
\RequirePackage{glossary-long}  
\RequirePackage{glossary-longragged}
```

(`longtable` and `array` loaded by those packages).

long-booktabs The `long-booktabs` style is similar to the `longheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

As with the `longheader` style, use the `long` style as a base.

```
\setglossarystyle{long}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname & \bfseries
\descriptionname\tabularnewline\midrule\endhead
\bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
```

long3col-booktabs The `long3col-booktabs` style is similar to the `long3colheader` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{long3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col` style as a base.

```
\setglossarystyle{long3col}%
```

Add a header with rules.

```
\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname &
\bfseries \descriptionname &
\bfseries \pagelistname
\tabularnewline\midrule\endhead
\bottomrule\endfoot}%
```

Check for the `nogroupskip` package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for `nogroupskip` should occur outside `\glsgroupskip` to be on the safe side.

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
```

```

\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

long4col-booktabs The long4col-booktabs style is similar to the long4colheader style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{long4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the long4col style as a base.

```
\setglossarystyle{long4col}%
```

Add a header with rules.

```

\renewcommand*{\glossaryheader}{%
\toprule \bfseries \entryname &
\bfseries \descriptionname &
\bfseries \symbolname &
\bfseries \pagelistname
\tabularnewline\midrule\endhead
\bottomrule\endfoot}%

```

Check for the nogroupskip package option. If there should be a gap between groups, insert the penalty and the vertical space. The check for nogroupskip should occur outside \glsgroupskip to be on the safe side.

```

\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}

```

altlong4col-booktabs The altlong4col-booktabs style is similar to the altlong4colheader style but uses the booktabs rules and patches longtable to check for group skip occurring at a page break.

```
\newglossarystyle{altlong4col-booktabs}{%
```

The patch \glspatchLToutput is already applied in long4col-booktabs and so doesn't need to be here.

```
\glspatchLToutput
```

Use the long4col-booktabs style as a base.

```
\setglossarystyle{long4col-booktabs}%
```

Change the column specifications:

```

\renewenvironment{theglossary}%
{\begin{longtable}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
{\end{longtable}}%
}

```

Ragged styles.

longragged-booktabs The `longragged-booktabs` style is similar to the `longragged` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long-booktabs` style as a base.

```
\setglossarystyle{long-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%  
  {\end{longtable}}%  
}
```

longragged3col-booktabs The `longragged3col-booktabs` style is similar to the `longragged3col` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{longragged3col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `long3col-booktabs` style as a base.

```
\setglossarystyle{long3col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%  
  {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}%  
    >{\raggedright}p{\glspagelistwidth}}}%  
  {\end{longtable}}%  
}
```

altlongragged4col-booktabs The `altlongragged4col-booktabs` style is similar to the `altlongragged4col` style but uses the `booktabs` rules and patches `longtable` to check for group skip occurring at a page break.

```
\newglossarystyle{altlongragged4col-booktabs}{%
```

If the style change is scoped, the patch will only have a local effect, which may be useful if it conflicts with other tables in the document.

```
\glspatchLToutput
```

Use the `altlong4col-booktabs` style as a base.

```
\setglossarystyle{altlong4col-booktabs}%
```

Adjust the column specification.

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

```
}
```

`\glsLTpenaltycheck`

```
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
```

`\glspenaltygroupskip`

```
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
```

`\glsrestoreLToutput` Provide a way of restoring `\LT@output` for the user.

```
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
```

This is David's patch, but I've replaced the hard-coded values with `\glsLTpenaltycheck` to make it easier to adjust.

`\glspatchLToutput`

```
\newcommand*{\glspatchLToutput}{%
\renewcommand*{\LT@output}{%
  \ifnum\outputpenalty <-\@Mi
    \ifnum\outputpenalty > -\LT@end@pen
      \LT@err{floats and marginpars not allowed in a longtable}\@ehc
    \else
      \setbox\z@\vbox{\unvbox\@cclv}%
      \ifdim \ht\LT@lastfoot>\ht\LT@foot
        \dimen@pagegoal
        \advance\dimen@-\ht\LT@lastfoot
        \ifdim\dimen@<\ht\z@
          \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
          \@makecol
          \@outputpage
          \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
        \fi
      \fi
      \global\@colroom\@colht
      \global\vsizel\@colht
      {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
    \fi
  \else
    \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
    \@makecol
    \@outputpage
```



```

\global\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{v5.1}{2026-04-21}

```

Declare package:

```
\ProvidesPackage{glossary-longragged}[2026/04/21 v5.1 (NLCT)]
```

Requires the `array` package:

```
\RequirePackage{array}
```

Requires the `longtable` package:

```
\RequirePackage{longtable}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may have already been defined.

```

\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}

```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}

```

`longragged` The `longragged` glossary style is like the `long` but uses ragged right formatting for the description column.

```
\newglossarystyle{longragged}{%
```

Use `longtable` with two columns:

```

\renewenvironment{theglossary}{%
  {\begin{longtable}{1>{\raggedright}p{glsdescwidth}}}%
  {\end{longtable}}}%

```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No heading between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries displayed in a row:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2%
  \tabularnewline
}%
```

Sub entries displayed on the following row without the name:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}%
  \glspostdescription\space ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
```

longraggedborder The `longraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{longraggedborder}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Use `longtable` with two columns with vertical lines between each column:

```
\renewenvironment{theglossary}{%
  \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
  {\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

longraggedheader The `longraggedheader` style is like the `longragged` style but with a header:

```
\newglossarystyle{longraggedheader}{%
```

Base it on the `glostylelongragged` style:

```
\setglossarystyle{longragged}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%  
  \bfseries \entryname & \bfseries \descriptionname  
  \tabularnewline\endhead}%  
}
```

longraggedheaderborder The longraggedheaderborder style is like the longragged style but with a header and border:

```
\newglossarystyle{longraggedheaderborder}{%
```

Base it on the glostylelongraggedborder style:

```
\setglossarystyle{longraggedborder}%
```

Set the table's header and add horizontal line to table's foot:

```
\renewcommand*{\glossaryheader}{%  
  \hline\bfseries \entryname & \bfseries \descriptionname  
  \tabularnewline\hline  
  \endhead  
  \hline\endfoot}%  
}
```

longragged3col The longragged3col style is like longragged but with 3 columns

```
\newglossarystyle{longragged3col}{%
```

Use a longtable with 3 columns:

```
\renewenvironment{theglossary}%  
  {\begin{longtable}[1>{\raggedright}p{\glsdescwidth}%  
    >{\raggedright}p{\glspagelistwidth}}}%  
  {\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No headings between groups:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%  
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} & ##2\tabularnewline  
}%
```

Sub-entries on a separate row (no name, description in second column, page list in third column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`longragged3colborder` The `longragged3colborder` style is like the `longragged3col` style but with a border:

```
\newglossarystyle{longragged3colborder}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Use a `longtable` with 3 columns with vertical lines around them:

```
\renewenvironment{theglossary}%
{\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
{\end{longtable}}%
```

Place horizontal lines at the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`longragged3colheader` The `longragged3colheader` style is like `longragged3col` but with a header row:

```
\newglossarystyle{longragged3colheader}{%
```

Base it on the `glostylelongragged3col` style:

```
\setglossarystyle{longragged3col}%
```

Set the table's header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`longragged3colheaderborder` The `longragged3colheaderborder` style is like the above but with a border

```
\newglossarystyle{longragged3colheaderborder}{%
```

Base it on the `glostylelongragged3colborder` style:

```
\setglossarystyle{longragged3colborder}%
```

Set the table's header and add horizontal line at table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

`altlongragged4col` The `altlongragged4col` style is like the `altlong4col` style defined in the `glossary-long` package, except that ragged right formatting is used for the description and page list columns.

```
\newglossarystyle{altlongragged4col}{%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
{\begin{longtable}[1>{\raggedright}p{\glsdescwidth}1%
>{\raggedright}p{\glspagerlistwidth}}}%
{\end{longtable}}%
```

No table header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a single row (name in first column, description in second column, symbol in third column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} & \glossentrysymbol{##1} &
##2\tabularnewline
}%
```

Sub entries on a single row with no name (description in second column, symbol in third column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
```

`altlongragged4colheader` The `altlongragged4colheader` style is like `altlongragged4col` but with a header row.

```
\newglossarystyle{altlongragged4colheader}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
```

Table has a header:

```
\renewcommand*{\glossaryheader}{%
  \bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\endhead}%
}
```

`altlongragged4colborder` The `altlongragged4colborder` style is like `altlongragged4col` but with a border.

```
\newglossarystyle{altlongragged4colborder}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
    >{\raggedright}p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Add horizontal lines to the head and foot of the table:

```
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
```

`altlongragged4colheaderborder` The `altlongragged4colheaderborder` style is like the above but with a header as well as a border.

```
\newglossarystyle{altlongragged4colheaderborder}{%
```

Base it on the `glostylealtlongragged4col` style:

```
\setglossarystyle{altlongragged4col}%
```

Use a `longtable` with 4 columns where the second and last columns may have multiple lines in each row:

```
\renewenvironment{theglossary}%
  {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|%
    >{\raggedright}p{\glspagelistwidth}|}%
  {\end{longtable}}%
```

Add table header and horizontal line at the table's foot:

```
\renewcommand*{\glossaryheader}{%
  \hline\bfseries\entryname&\bfseries\descriptionname&
  \bfseries \symbolname&
  \bfseries\pagelistname\tabularnewline\hline\endhead
  \hline\endfoot}%
}
```

3.7 Glossary Styles using multicol (glossary-mcols.sty)

The glossary-mcols style file defines glossary styles that use the multicol package. These use the tree-like glossary styles in a multicol environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-mcols-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-mcols-2021-11-01.sty}
\DeclareRelease{v4.58}{2025-03-19}{glossary-mcols-2025-03-19.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-mcols}[2026/04/21 v5.1 (NLCT)]
```

Required packages:

```
\RequirePackage{multicol}
\RequirePackage{glossary-tree}
\ExplSyntaxOn
```

```
\GlsMcolTreeSetup
```

```
\NewDocumentCommand \GlsMcolTreeSetup { m }
{
  \keys_set:nn { glossaries / styles / mcoltree * } { #1 }
}
\keys_define:nn { glossaries / styles }
{
  mcoltree* .code:n =
  {
    \GlsMcolTreeSetup { #1 }
  } ,
}
```

mcoltree* Version 4.59: new glossary style with flexible $\langle key \rangle = \langle value \rangle$ options. This has its own formatting commands.

```
\tl_new:N \l__glossaries_mcols_env_tl
\tl_set:Nn \l__glossaries_mcols_env_tl { multicol }
\tl_new:N \l__glossaries_mcols_envopt_tl
\int_new:N \l__glossaries_mcols_columns_int
\bool_new:N \l__glossaries_mcols_spannav_bool
\keys_define:nn { glossaries / styles / mcoltree * }
{
  columns .int_set:N = \l__glossaries_mcols_columns_int ,
  columns .initial:n = 2 ,
  balance .choice: ,
  balance / true .code:n =
  {
    \tl_set:Nn \l__glossaries_mcols_env_tl { multicol }
  } ,
  balance / false .code:n =
```

```

    {
      \tl_set:Nn \l__glossaries_mcols_env_tl { multicol * }
    } ,
    balance .default:n = true ,
    span-nav .choice: ,
    span-nav / true .code:n =
    {
      \bool_set_true:N \l__glossaries_mcols_spannav_bool
      \tl_set:Nn \l__glossaries_mcols_envopt_tl
      {
        \bool_if:NT \l__glossaries_tree_hypernav_bool
        {
          [ \exp_not:N \__glossaries_tree_header: ]
        }
      }
    } ,
    span-nav / false .code:n =
    {
      \bool_set_false:N \l__glossaries_mcols_spannav_bool
      \tl_set:Nn \l__glossaries_mcols_envopt_tl { \relax }
    } ,
    span-nav .default:n = true ,
    span-nav .initial:n = true ,
  }
\newglossarystyle{mcoltree*}
{
  \setglossarystyle { tree* }
  \renewenvironment { theglossary }
  {
    \exp_last_unbraced:Ne \__glossaries_mcols_begin:
    \l__glossaries_mcols_envopt_tl
    \__glossaries_tree_begin:
  }
  {
    \__glossaries_tree_end:
    \__glossaries_mcols_end:
  }
  \renewcommand \glossaryheader
  {
    \bool_if:NF \l__glossaries_mcols_spannav_bool
    {
      \__glossaries_tree_header:
    }
  }
}
\cs_new:Nn \__glossaries_mcols_begin:
{

```



```

\exp_args:NVV \begin
  \l__glossaries_mcols_env_tl
  \l__glossaries_mcols_columns_int
}
\cs_new:Nn \__glossaries_mcols_end:
{
  \exp_args:NV \end
  \l__glossaries_mcols_env_tl
}

\ExplSyntaxOff

```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```

\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}

```

`\glsmcols` Define macro in which to store the number of columns. (Defaults to 2.)

```

\newcommand*{\glsmcols}{2}

```

`mcolindex` Multi-column index style. Same as the index, but puts the glossary in multiple columns. (Ideally the glossary title should go in the optional argument of `multicols`, but the title isn't part of the glossary style.)

```

\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%

    \let\item\glstreeitem
    \let\subitem\glstreesubitem
    \let\subsubitem\glstreesubsubitem
  }%
  {\end{multicols}}%
}

```

`mcolindexgroup` As `mcolindex` but has headings:

```

\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
}

```

`mcolindexhypergroup` The `mcolindexhypergroup` style is like the `mcolindexgroup` style but has hyper navigation.

```

\newglossarystyle{mcolindexhypergroup}{%

```

Base it on the `glostylemcolindex` style:

```
\setglossarystyle{mcolindex}%
```

Put navigation links to the groups at the start of the glossary:

```
\renewcommand*{\glossaryheader}{%
  \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
```

`mcolindexspannav` Similar to `mcolindexhypergroup`, but puts the navigation line in the optional argument of `multicols`.

```
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
  {%

    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%

    \let\item\glstreeitem}%
  {\end{multicols}}%
```

Add a heading for each group (with a target). The group's title is in bold followed by a vertical gap.

```
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
```

`mcoltree` Multi-column index style. Same as the tree, but puts the glossary in multiple columns.

```
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%

    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
```

`mcoltreegroup` Like the `mcoltree` style but the glossary groups have headings.

```
\newglossarystyle{mcoltreegroup}{%
```

Base it on the `glostylemcoltree` style:

```
\setglossarystyle{mcoltree}%
```

Each group has a heading (in bold) followed by a vertical gap):

```
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
```

`mcoltreehypergroup` The `mcoltreehypergroup` style is like the `treegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcoltreehypergroup}{%
```

Base it on the `glostylemcoltree` style:

```
\setglossarystyle{mcoltree}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreespannav` Similar to the `mcoltreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcoltreespannav}{%
```

```
\setglossarystyle{tree}%
```

```
\renewenvironment{theglossary}%
```

```
{%
```

```
\begin{multicols}{\glsmcols}[ \noindent\glstreenavigationfmt{\glsnavigation}]
```

```
\setlength{\parindent}{0pt}%
```

```
\setlength{\parskip}{0pt plus 0.3pt}%
```

```
}%
```

```
{\end{multicols}}}%
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

`mcoltreenoname` Multi-column index style. Same as the `treenoname`, but puts the glossary in multiple columns.

```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}

```

mcoltreenonamegroup Like the **mcoltreenoname** style but the glossary groups have headings.

```

\newglossarystyle{mcoltreenonamegroup}{%
  Base it on the glostylemcoltreenoname style:
  \setglossarystyle{mcoltreenoname}%
  Give each group a heading:
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}

```

mcoltreenonamehypergroup The **mcoltreenonamehypergroup** style is like the **mcoltreenonamegroup** style, but has a set of links to the groups at the start of the glossary.

```

\newglossarystyle{mcoltreenonamehypergroup}{%
  Base it on the glostylemcoltreenoname style:
  \setglossarystyle{mcoltreenoname}%
  Put navigation links to the groups at the start of the theglossary environment:
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  Each group has a heading (in bold with a target) followed by a vertical gap):
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}

```

mcoltreenonamespannav Similar to the **mcoltreenonamehypergroup** style but the navigation line is put in the optional argument of the multicols environment.

```

\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}

```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
```

mcolalttree Multi-column index style. Same as the **alttree**, but puts the glossary in multiple columns.

```
\newglossarystyle{mcolalttree}{%
\setglossarystyle{alttree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
}
```

mcolalttreegroup Like the **mcolalttree** style but the glossary groups have headings.

```
\newglossarystyle{mcolalttreegroup}{%
```

Base it on the **glostylemcolalttree** style:

```
\setglossarystyle{mcolalttree}%
```

Give each group a heading.

```
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
```

mcolalttreehypergroup The **mcolalttreehypergroup** style is like the **mcolalttreegroup** style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{mcolalttreehypergroup}{%
```

Base it on the **glostylemcolalttree** style:

```
\setglossarystyle{mcolalttree}%
```

Put the navigation links in the header

```
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glssnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

`mcolalttreespannav` Similar to the `mcolalttreehypergroup` style but the navigation line is put in the optional argument of the `multicols` environment.

```
\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glssnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
```

Put a hypertarget at the start of each group

```
\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glssnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}
```

3.8 Glossary Styles using supertabular environment (glossary-super.sty)

The glossary styles defined in the `glossary-super` package use the `supertabular` environment.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-super-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossary-super-2021-11-01.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-super}[2026/04/21 v5.1 (NLCT)]
```

Requires the `supertabular` package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined if `glossary-long` has been loaded.

```
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined if `glossary-long` has been loaded.

```
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{}
```

super The `super` glossary style uses the `supertabular` environment (it uses lengths defined in the `glossary-long` package.)

```
\newglossarystyle{super}{%
```

Put the glossary in a `supertabular` environment with two columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}}\tabletail{}%
  \begin{supertabular}{lp{glsdescwidth}}%
  {\end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%
  \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
```

```

        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& \tabularnewline}%
    \fi
}

```

superborder The **superborder** style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superborder}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|p{\glsdescwidth}|}%
\end{supertabular}}%
}

```

superheader The **superheader** style is like the **super** style, but with a header:

```
\newglossarystyle{superheader}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns, a header and no tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\bfseries \entryname &
\bfseries \descriptionname \tabularnewline}%
\tabletail{}%
\begin{supertabular}{lp{\glsdescwidth}}%
\end{supertabular}}%
}

```

superheaderborder The **superheaderborder** style is like the **super** style but with a header and border:

```
\newglossarystyle{superheaderborder}{%
```

Base it on the **glostylesuper** style:

```
\setglossarystyle{super}%
```

Put the glossary in a **supertabular** environment with two columns, a header and horizontal lines above and below the table:

```

\renewenvironment{theglossary}%
{
\tablehead{\hline\bfseries \entryname &
\bfseries \descriptionname \tabularnewline\hline}%
\tabletail{\hline}
\begin{supertabular}{|l|p{\glsdescwidth}|}%
\end{supertabular}}%
}

```


super3col The `super3col` style is like the `super` style, but with 3 columns:

```
\newglossarystyle{super3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{}\tabletail{}}%  
  {\begin{supertabular}{lp{\glstdescwidth}p{\glspagelistwidth}}}%  
  {\end{supertabular}}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgrouphheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%  
  \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1} & ##2\tabularnewline  
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%  
  &  
  \glssubentryitem{##2}%  
  \glstarget{##2}{\strut}\glossentrydesc{##2} &  
  ##3\tabularnewline  
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip  
  \renewcommand*{\glsgroupskip}{}%  
\else  
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%  
\fi  
}
```

super3colborder The `super3colborder` style is like the `super3col` style, but with a border:

```
\newglossarystyle{super3colborder}{%
```

Base it on the `glostylesuper3col` style:

```
\setglossarystyle{super3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%  
  {\tablehead{\hline}\tabletail{\hline}}%
```

```

        \begin{supertabular}{|l|p{\glsgdescwidth}|p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    }

```

super3colheader The `super3colheader` style is like the `super3col` style but with a header row:

```

    \newglossarystyle{super3colheader}{%
Base it on the glostylesuper3col style:
    \setglossarystyle{super3col}%
Put the glossary in a supertabular environment with three columns, a header
and no tail:
    \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    \begin{supertabular}{|p{\glsgdescwidth}p{\glspagelistwidth}|}%
    {\end{supertabular}}%
    }

```

super3colheaderborder The `super3colheaderborder` style is like the `super3col` style but with a header and border:

```

    \newglossarystyle{super3colheaderborder}{%
Base it on the glostylesuper3colborder style:
    \setglossarystyle{super3colborder}%
Put the glossary in a supertabular environment with three columns, a header
with horizontal lines and a horizontal line in the tail:
    \renewenvironment{theglossary}%
    {\tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|l|p{\glsgdescwidth}|p{\glspagelistwidth}|}%
    {\end{supertabular}}%
    }

```

super4col The `super4col` glossary style has four columns, where the third column contains the value of the corresponding symbol key used when that entry was defined.

```

    \newglossarystyle{super4col}{%
Put the glossary in a supertabular environment with four columns and no head
or tail:
    \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    \begin{supertabular}{|l|l|l|l|}%
    \end{supertabular}%
Do nothing at the start of the table:
    \renewcommand*{\glossaryheader}{}%

```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

super4colheader The `super4colheader` style is like the `super4col` but with a header row.

```
\newglossarystyle{super4colheader}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l1l1l}%
  {\end{supertabular}}%
}
```

super4colborder The `super4colborder` style is like the `super4col` but with a border.

```
\newglossarystyle{super4colborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
   \begin{supertabular}{|l|l|l|l|}%
   {\end{supertabular}}%
```

super4colheaderborder The `super4colheaderborder` style is like the `super4col` but with a header and border.

```
\newglossarystyle{super4colheaderborder}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
   \tabletail{\hline}%
   \begin{supertabular}{|l|l|l|l|}%
   {\end{supertabular}}%
```

altsuper4col The `altsuper4col` glossary style is like `super4col` but has provision for multiline descriptions.

```
\newglossarystyle{altsuper4col}{%
```

Base it on the `glostylesuper4col` style:

```
\setglossarystyle{super4col}%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
  {\tablehead{}\tabletail{}}%
  \begin{supertabular}{lp{\glsdescwidth}lp{\glspagelistwidth}}%
  {\end{supertabular}}%
```

altsuper4colheader The `altsuper4colheader` style is like the `altsuper4col` but with a header row.

```
\newglossarystyle{altsuper4colheader}{%
```

Base it on the `glostylesuper4colheader` style:

```
\setglossarystyle{super4colheader}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\tabletail{}%
        \begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
        {\end{supertabular}}%
    }

```

altsuper4colborder The `altsuper4colborder` style is like the `altsuper4col` but with a border.

```
\newglossarystyle{altsuper4colborder}{%
```

Base it on the `glostylesuper4colborder` style:

```
\setglossarystyle{super4colborder}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
    {\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}

```

altsuper4colheaderborder The `altsuper4colheaderborder` style is like the `altsuper4col` but with a header and border.

```
\newglossarystyle{altsuper4colheaderborder}{%
```

Base it on the `glostylesuper4colheaderborder` style:

```
\setglossarystyle{super4colheaderborder}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```

    \renewenvironment{theglossary}%
    {\tablehead{\hline
        \bfseries\entryname &
        \bfseries\descriptionname &
        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
    {\lllp{\glsgdescwidth}\lllp{\glspagelistwidth}}}%
    {\end{supertabular}}%
}

```

3.9 Glossary Styles using supertabular environment (glossary-superragged.sty)

The glossary styles defined in the `glossary-superragged` package use the `supertabular` environment. These styles are like those provided by the `glossary-super` package, except that the multiline columns have ragged right justification.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```
\DeclareRelease{v4.46}{2020-03-19}{glossary-superragged-2020-03-19.sty}  
\DeclareRelease{v4.49}{2021-11-01}{glossary-superragged-2021-11-01.sty}  
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-superragged}[2026/04/21 v5.1 (NLCT)]
```

Requires the array package:

```
\RequirePackage{array}
```

Requires the supertabular package:

```
\RequirePackage{supertabular}
```

`\glsdescwidth` This is a length that governs the width of the description column. This may already have been defined.

```
\@ifundefined{glsdescwidth}{%  
  \newlength{glsdescwidth}  
  \setlength{glsdescwidth}{0.6\hsize}  
}{}
```

`\glspagelistwidth` This is a length that governs the width of the page list column. This may already have been defined.

```
\@ifundefined{glspagelistwidth}{%  
  \newlength{glspagelistwidth}  
  \setlength{glspagelistwidth}{0.1\hsize}  
}{}
```

`superragged` The `superragged` glossary style uses the `supertabular` environment.

```
\newglossarystyle{superragged}{%
```

Put the glossary in a `supertabular` environment with two columns and no head or tail:

```
\renewenvironment{theglossary}{%  
  {\tablehead}{\tabletail}{%  
    \begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%  
  {\end{supertabular}}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries put in a row (name in first column, description and page list in second column):

```
\renewcommand{\glossentry}[2]{%  
  \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &  
  \glossentrydesc{##1}\glspostdescription\space ##2%  
  \tabularnewline  
}%
```

Sub entries put in a row (no name, description and page list in second column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
  ##3%
  \tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& \tabularnewline}%
\fi
}
```

superraggedborder The `superraggedborder` style is like the above, but with horizontal and vertical lines:

```
\newglossarystyle{superraggedborder}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>\raggedright}p{\glsdescwidth}|}%
  {\end{supertabular}}%
```

superraggedheader The `superraggedheader` style is like the `super` style, but with a header:

```
\newglossarystyle{superraggedheader}{%
```

Base it on the `glostylesuperragged` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries \entryname & \bfseries \descriptionname
  \tabularnewline}%
  \tabletail{}%
  \begin{supertabular}{l>\raggedright}p{\glsdescwidth}}%
  {\end{supertabular}}%
```

superraggedheaderborder The `superraggedheaderborder` style is like the `superragged` style but with a header and border:

```
\newglossarystyle{superraggedheaderborder}{%
```

Base it on the `glostylessuper` style:

```
\setglossarystyle{superragged}%
```

Put the glossary in a `supertabular` environment with two columns, a header and horizontal lines above and below the table:

```
\renewenvironment{theglossary}%
{
  \tablehead{\hline\bfseries \entryname &
    \bfseries \descriptionname\tabularnewline\hline}%
  \tabletail{\hline}
  \begin{supertabular}{|l|>{\raggedright}p{\glstdescwidth}}
  \end{supertabular}}%
```

```
}
```

superragged3col The `superragged3col` style is like the `superragged` style, but with 3 columns:

```
\newglossarystyle{superragged3col}{%
```

Put the glossary in a `supertabular` environment with three columns and no head or tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{}\tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glstdescwidth}%
    >{\raggedright}p{\glspagelistwidth}}
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row (name in first column, description in second column, page list in last column):

```
\renewcommand{\glossentry}[2]{%
  \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  ##2\tabularnewline
}%
```

Sub entries on a row (no name, description in second column, page list in last column):

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  ##3\tabularnewline
}%
```


Blank row between groups: The check for nogroupskip must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```
\ifglsnogroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
```

superragged3colborder The `superragged3colborder` style is like the `superragged3col` style, but with a border:

```
\newglossarystyle{superragged3colborder}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns and a horizontal line in the head and tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\hline}\tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

superragged3colheader The `superragged3colheader` style is like the `superragged3col` style but with a header row:

```
\newglossarystyle{superragged3colheader}{%
```

Base it on the `glostylesuperragged3col` style:

```
\setglossarystyle{superragged3col}%
```

Put the glossary in a `supertabular` environment with three columns, a header and no tail:

```
\renewenvironment{theglossary}%
  {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline}\tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}%
    >{\raggedright}p{\glspagelistwidth}|}%
  {\end{supertabular}}%
}
```

raggedright3colheaderborder The `superragged3colheaderborder` style is like the `superragged3col` style but with a header and border:

```
\newglossarystyle{superragged3colheaderborder}{%
```

Base it on the `glostylesuperragged3colborder` style:

```
\setglossarystyle{superragged3colborder}%
```

Put the glossary in a `supertabular` environment with three columns, a header with horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{\hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}|%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
}
```

`altsuperragged4col` The `altsuperragged4col` glossary style is like `altsuper4col` style in the `glossary-super` package but uses ragged right formatting in the description and page list columns.

```
\newglossarystyle{altsuperragged4col}{%
```

Put the glossary in a `supertabular` environment with four columns and no head or tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{} \tabletail{}%
  \begin{supertabular}{|l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}|}%
  \end{supertabular}}%
```

Do nothing at the start of the table:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries on a row with the name in the first column, description in second column, symbol in third column and page list in last column:

```
\renewcommand{\glossentry}[2]{%
  \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
  \glossentrydesc{##1} &
  \glossentrysymbol{##1} & ##2\tabularnewline
}%
```

Sub entries on a row with no name, the description in the second column, symbol in third column and page list in last column:

```
\renewcommand{\subglossentry}[3]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}\glossentrydesc{##2} &
  \glossentrysymbol{##2} & ##3\tabularnewline
}%
```

Blank row between groups: The check for `nogroupskip` must occur outside `\glsgroupskip` (<http://www.dickimaw-books.com/cgi-bin/bugtracker.cgi?action=view&key=108>)

```

\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & & \tabularnewline}%
\fi
}

```

altsuperragged4colheader The `altsuperragged4colheader` style is like the `altsuperragged4col` style but with a header row.

```
\newglossarystyle{altsuperragged4colheader}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns, a header and no tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\symbolname &
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{1>\raggedright}p{\glsgdescwidth}l%
>\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
}

```

altsuperragged4colborder The `altsuperragged4colborder` style is like the `altsuperragged4col` style but with a border.

```
\newglossarystyle{altsuperragged4colborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuper4col}%
```

Put the glossary in a `supertabular` environment with four columns and a horizontal line in the head and tail:

```

\renewenvironment{theglossary}%
{
\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}%
{
|l|>\raggedright}p{\glsgdescwidth}l|l|%
>\raggedright}p{\glspagelistwidth}l|l|}%
{\end{supertabular}}%
}

```

superragged4colheaderborder The `altsuperragged4colheaderborder` style is like the `altsuperragged4col` style but with a header and border.

```
\newglossarystyle{altsuperragged4colheaderborder}{%
```

Base it on the `glostylealtsuperragged4col` style:

```
\setglossarystyle{altsuperragged4col}%
```

Put the glossary in a `supertabular` environment with four columns and a header bordered by horizontal lines and a horizontal line in the tail:

```
\renewenvironment{theglossary}%
{
  \tablehead{\hline
    \bfseries\entryname &
    \bfseries\descriptionname &
    \bfseries\symbolname &
    \bfseries\pagelistname\tabularnewline\hline}%
  \tabletail{\hline}%
  \begin{supertabular}%
    {
      |l|>{\raggedright}p{\glsdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}%
  }
\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}
\DeclareRelease{v4.58}{2025-03-19}{glossary-tree-2025-03-19.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}
```

Declare package:

```
\ProvidesPackage{glossary-tree}[2026/04/21 v5.1 (NLCT)]
```

tree* Version 4.59: new glossary style with flexible $\langle key \rangle = \langle value \rangle$ options. This has its own formatting commands.

```
\ExplSyntaxOn
```

User-level commands:

```
\GlsTreeSetup

\NewDocumentCommand \GlsTreeSetup { m }
{
  \keys_set:nn { glossaries / styles / tree * } { #1 }
}
\keys_define:nn { glossaries / styles }
{
  tree* .code:n =
  {
    \GlsTreeSetup { #1 }
  } ,
}
}
```

`\GlsTreeUpdateWidestNameOrSymbol[⟨level⟩]{⟨label⟩}`

TreeUpdateWidestNameOrSymbol

```
\NewDocumentCommand \GlsTreeUpdateWidestNameOrSymbol { o m }
{
  \IfValueTF { #1 }
  {
    \glossaries_tree_update_widest_name:nn
    { #1 }
    { \glentryname { #2 } }
    \glossaries_tree_update_widest_symbol:nn
    { #1 }
    { \glentrysymbol { #2 } }
  }
  {
    \exp_args:Nv
    \glossaries_tree_update_widest_name:nn
    { glo@ \glsdetoklabel { #2 } @level }
    { \glentryname { #2 } }
    \exp_args:Nv
    \glossaries_tree_update_widest_symbol:nn
    { glo@ \glsdetoklabel { #2 } @level }
    { \glentrysymbol { #2 } }
  }
}
```

`\GlsTreeUpdateWidestNameOrSymbol[⟨level⟩]{⟨label⟩}`

TreeUpdateWidestNameAndSymbol

```
\NewDocumentCommand \GlsTreeUpdateWidestNameAndSymbol { o m }
{
  \IfValueTF { #1 }
  {
    \glossaries_tree_update_widest_name_symbol:nnn
    { #1 }
    { \glentryname { #2 } }
    { \glentrysymbol { #2 } }
  }
  {
    \exp_args:Nv
    \glossaries_tree_update_widest_name_symbol:nnn
    { glo@ \glsdetoklabel { #2 } @level }
    { \glentryname { #2 } }
    { \glentrysymbol { #2 } }
  }
}
```

`\GlsTreeStarBox`

```
\newcommand\GlsTreeStarBox[1]{#1}
```

```
\GlsTreeStarOuterBox
\newcommand\GlsTreeStarOuterBox[1]{\GlsTreeStarBox{#1}}
```

```
\GlsTreeStarNameBox
\newcommand\GlsTreeStarNameBox[1]{\GlsTreeStarBox{#1}}
```

```
\GlsTreeStarSymbolBox
\newcommand\GlsTreeStarSymbolBox[1]{\GlsTreeStarBox{#1}}
```

```
\GlsTreeStarItemCounterBox
\newcommand\GlsTreeStarItemCounterBox[1]{#1}
```

```
\GlsTreeStarSubItemCounterBox
\newcommand\GlsTreeStarSubItemCounterBox[1]{
\GlsTreeStarItemCounterBox { #1 }
}
```

```
\glossaries_tree_bookmark_group:nnn {\langle level \rangle} {\langle label \rangle}
{\langle title \rangle}
```

Used in hook before group header to bookmark the group (only if header shown).

```
\cs_if_exist:NTF \pdfbookmark
{
\cs_new:Nn \glossaries_tree_bookmark_group:nnn
{
\pdfbookmark [ #1 ] { #3 } { \currentglossary . #2 }
}
}
{
\cs_new:Nn \glossaries_tree_bookmark_group:nnn { }
}
```

```
\glossaries_tree_subgroup_title:nn {\langle parent-label \rangle}
{\langle group title \rangle}
```

Sub-group header title.

```
\cs_new:Nn \glossaries_tree_subgroup_title:nn { #2 }
```

Version 4.6: hook before sub-group (label and level):

```
\cs_new:Nn \glossaries_tree_pre_subgroup:nn
{
\glsgroupskip
}
```

Hook used at the start of each item (arguments: level, label, location list).

```
\cs_new:Nn \glossaries_tree_pre_item:nnn { }
```

Hook used at the end of each item (arguments: level, label, location list).

```
\cs_new:Nn \glossaries_tree_post_item:nnn { }
```

Natural width doesn't require a box but does need scoping to localise font changes.

```
\cs_new:Nn \glossaries_tree_namenobox:n
{ \GlsTreeStarNameBox { { #1 } } }
\cs_new:Nn \glossaries_tree_namebox:nnn
{
  \GlsTreeStarNameBox { \makebox [ #1 ] [ #2 ] { #3 } }
}
\cs_generate_variant:Nn \glossaries_tree_namebox:nnn { vVn , vvn }
```

Similarly for symbol.

```
\cs_new:Nn \glossaries_tree_symbolnobox:n
{ \GlsTreeStarSymbolBox { { #1 } } }
\cs_new:Nn \glossaries_tree_symbolbox:nnn
{
  \GlsTreeStarSymbolBox { \makebox [ #1 ] [ #2 ] { #3 } }
}
\cs_generate_variant:Nn \glossaries_tree_symbolbox:nnn
{ vVn , vvn }
```

Encapsulates both name and symbol:

```
\cs_new:Nn \glossaries_tree_namesymbolnobox:n
{ \GlsTreeStarOuterBox { #1 } }
\cs_new:Nn \glossaries_tree_namesymbolbox:nnn
{
  \GlsTreeStarOuterBox { \makebox [ #1 ] [ #2 ] { #3 } }
}
\cs_generate_variant:Nn \glossaries_tree_namesymbolbox:nnn
{ vVn , vvn , nVn , nvn }
```

Parentheses:

```
\cs_new:Nn \glossaries_tree_paren:n { ( #1 ) }
```

For the top-level and level 1 entry item counter:

```
\cs_new:Nn \glossaries_tree_entryitem_nobox:n
{
  \GlsTreeStarItemCounterBox { #1 }
}
\cs_new:Nn \glossaries_tree_entryitem_box:nnn
{
  \GlsTreeStarItemCounterBox { \makebox [ #1 ] [ #2 ] { #3 } }
}
\cs_generate_variant:Nn \glossaries_tree_entryitem_box:nnn { VVn }
\cs_new:Nn \glossaries_tree_subentryitem_nobox:n
{
  \GlsTreeStarSubItemCounterBox { { #1 } }
}
\cs_new:Nn \glossaries_tree_subentryitem_box:nnn
```

```

{
  \GlsTreeStarSubItemCounterBox { \makebox [ #1 ] [ #2 ] { #3 } }
}
\cs_generate_variant:Nn \glossaries_tree_subentryitem_box:nnn { VVn }
Clear or zero all widest name and symbol variables.
\cs_new:Nn \glossaries_tree_reset_all_widest:
{
  \int_step_inline:nnn
  { \c_zero_int } { l__glossaries_tree_max_known_level_int }
  {
    \exp_args:Ne \__glossaries_tree_reset_all_widest:n
    { \romannumeral ##1 }
  }
}
\cs_new:Nn \__glossaries_tree_reset_all_widest:n
{
  \tl_if_exist:cT
  { l__glossaries_tree_namewidth #1 _tl }
  {
    \tl_set:cn
    { l__glossaries_tree_namewidth #1 _tl }
    { \c_zero_dim }
  }
  \tl_if_exist:cT
  { l__glossaries_tree_symbolwidth #1 _tl }
  {
    \tl_set:cn
    { l__glossaries_tree_symbolwidth #1 _tl }
    { \c_zero_dim }
  }
  \tl_if_exist:cT { @glswidestname #1 }
  {
    \tl_clear:c { @glswidestname #1 }
  }
  \tl_if_exist:cT { l__glossaries_tree_widest_symbol #1 _tl }
  {
    \tl_clear:c { l__glossaries_tree_widest_symbol #1 _tl }
  }
  \dim_if_exist:cT { l__glossaries_tree_namesymonlywidth #1 _dim }
  {
    \dim_zero:c { l__glossaries_tree_namesymonlywidth #1 _dim }
  }
  \tl_if_exist:cT { l__glossaries_tree_namesymbolwidth #1 _tl }
  {
    \tl_set:cn { l__glossaries_tree_namesymbolwidth #1 _tl }
    { \c_zero_dim }
  }
}

```

Setup name width for given level:


```

\cs_new:Nn \glossaries_tree_set_name_width:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \tl_clear_new:c
    { l__glossaries_tree_namewidth \romannumeral #1 _tl }
  \tl_if_eq:nnTF { #2 } { natural }
  {
    \tl_set:cn { l__glossaries_tree_namewidth \romannumeral #1 _tl }
      { \c_zero_dim }
  }
  {
    \tl_if_eq:nnF { #2 } { widest }
    {
      \tl_set:cn
        { l__glossaries_tree_namewidth \romannumeral #1 _tl }
        { #2 }
    }
  }
}

```

Setup symbol width for given level:

```

\cs_new:Nn \glossaries_tree_set_symbol_width:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \tl_clear_new:c
    { l__glossaries_tree_symbolwidth \romannumeral #1 _tl }
  \tl_if_eq:nnTF { #2 } { natural }
  {
    \tl_set:cn { l__glossaries_tree_symbolwidth \romannumeral #1 _tl }
      { \c_zero_dim }
  }
  {
    \tl_if_eq:nnF { #2 } { widest }
    {
      \tl_set:cn
        { l__glossaries_tree_symbolwidth \romannumeral #1 _tl }
        { #2 }
    }
  }
}

```

Identify the widest name. This uses the old internal command name for

backward-compatibility.

```
\cs_new:Nn \glossaries_tree_set_widest_name:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \tl_set:cn { @glswidestname \romannumeral #1 }
  { #2 }
}
```

Global:

```
\cs_new:Nn \glossaries_tree_gset_widest_name:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_gset:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \tl_gset:cn { @glswidestname \romannumeral #1 }
  { #2 }
}
```

Identify the widest symbol.

```
\cs_new:Nn \glossaries_tree_set_widest_symbol:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \tl_set:cn { l__glossaries_tree_widest_symbol \romannumeral #1 _tl }
  { #2 }
}
```

Global:

```
\cs_new:Nn \glossaries_tree_gset_widest_symbol:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_gset:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \tl_gset:cn { l__glossaries_tree_widest_symbol \romannumeral #1 _tl }
  { #2 }
}
```

Set the combined name and symbol width.

```
\cs_new:Nn \glossaries_tree_set_name_symbol_width:nn
{
```

```

\int_compare:nNnT
{ #1 } > { \l__glossaries_tree_max_known_level_int }
{
  \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
}
\tl_clear_new:c
{ \l__glossaries_tree_namesymbolwidth \romannumeral #1 _tl }
\tl_if_eq:nnTF { #2 } { natural }
{
  \tl_set:cn { \l__glossaries_tree_namesymbolwidth \romannumeral #1 _tl }
  { \c_zero_dim }
}
{
  \tl_if_eq:nnTF { #2 } { match ~ higher }
  {
    \tl_if_exist:cTF
    {
      \l__glossaries_tree_namesymbolwidth
      \romannumeral \int_eval:n { #1 - \c_one_int } _tl
    }
    {
      \tl_set:ce
      { \l__glossaries_tree_namesymbolwidth \romannumeral #1 _tl }
      {
        \exp_not:N \__glossaries_tree_match_namesymbolwidth:n
        { \romannumeral \int_eval:n { #1 - \c_one_int } }
      }
    }
  }
  {
    \PackageError { glossary-tree }
    {
      Can't ~ match ~ name-symbol-width ~ level ~ #1 ~
      to ~ level ~ \int_eval:n { #1 - \c_one_int }
    }
    {
      There's ~ either ~ no ~ higher ~ level ~ or ~
      no ~ width ~ has ~ been ~ set ~ for ~ that ~ level
    }
  }
}
}
{
  \tl_if_eq:nnF { #2 } { widest }
  {
    \tl_set:cn
    { \l__glossaries_tree_namesymbolwidth \romannumeral #1 _tl }
    { #2 }
  }
}
}
}

```

```

\cs_new:Nn \__glossaries_tree_match_namesymbolwidth:n
{
  \dim_use:c { l__glossaries_tree_namesymbolwidth #1 _dim }
}

```

Unlike the `glossaries-extra-stylemods \glupdatewidest` command, this doesn't re-measure the current widest name.

```

\cs_new:Nn \glossaries_tree_update_widest_name:nn
{
  \int_compare:nNnT
    { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \exp_args:Ne \__glossaries_tree_update_widest_name:nn
    { \romannumeral #1 } { #2 }
}
\cs_generate_variant:Nn \glossaries_tree_update_widest_name:nn
{ ne , nV , nv , no }
\cs_new:Nn \__glossaries_tree_update_widest_name:nn
{
  \dim_if_exist:cT { l__glossaries_tree_namesymonlywidth #1 _dim }
  {

```

Override combined name and symbol update.

```

    \dim_zero:c { l__glossaries_tree_namesymonlywidth #1 _dim }
  }
  \dim_if_exist:cF
  { l__glossaries_tree_namewidth #1 _dim }
  {
    \dim_new:c { l__glossaries_tree_namewidth #1 _dim }
  }
  \bool_lazy_or:nnTF
  {
    \bool_not_p:n
      { \tl_if_exist_p:c { l__glossaries_tree_namewidth #1 _tl } }
  }
  { \tl_if_empty_p:c { l__glossaries_tree_namewidth #1 _tl } }
  {
    \__glossaries_tree_init_namewidth:n { #1 }
  }
  {
    \dim_set:cn { l__glossaries_tree_namewidth #1 _dim }
    { \tl_use:c { l__glossaries_tree_namewidth #1 _tl } }
  }
  \tl_if_empty:nTF { #1 }
  {
    \__glossaries_tree_measure_name:Nn \l__glossaries_tmpa_dim { #2 }
  }
  {

```

```

    \__glossaries_tree_measure_child_name:Nn
    \l__glossaries_tmpa_dim { #2 }
  }
\dim_compare:nNnT
{ \l__glossaries_tmpa_dim } >
{ \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
{
  \dim_set_eq:cN { l__glossaries_tree_namewidth #1 _dim }
  \l__glossaries_tmpa_dim
  \tl_set:cn { @glswidestname #1 } { #2 }
  \tl_set:cn { l__glossaries_tree_namewidth #1 _tl }
  { \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
}
}

```

Similarly for the symbol:

```

\cs_new:Nn \glossaries_tree_update_widest_symbol:nn
{
  \int_compare:nNnT
  { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \exp_args:Ne \__glossaries_tree_update_widest_symbol:nn
  { \romannumeral #1 } { #2 }
}
\cs_generate_variant:Nn \glossaries_tree_update_widest_symbol:nn
{ ne , nV , nv , no }
\cs_new:Nn \__glossaries_tree_update_widest_symbol:nn
{
  \dim_if_exist:cT { l__glossaries_tree_namesymonlywidth #1 _dim }
  {

```

Override combined name and symbol update.

```

    \dim_zero:c { l__glossaries_tree_namesymonlywidth #1 _dim }
  }
  \dim_if_exist:cF
  { l__glossaries_tree_symbolwidth #1 _dim }
  {
    \dim_new:c { l__glossaries_tree_symbolwidth #1 _dim }
  }
  \bool_lazy_or:nnTF
  {
    \bool_not_p:n
    { \tl_if_exist_p:c { l__glossaries_tree_symbolwidth #1 _tl } }
  }
  { \tl_if_empty_p:c { l__glossaries_tree_symbolwidth #1 _tl } }
  {
    \__glossaries_tree_init_symbolwidth:n { #1 }
  }
}

```

```

\dim_set:cn { l__glossaries_tree_symbolwidth #1 _dim }
{ \tl_use:c { l__glossaries_tree_symbolwidth #1 _tl } }
}
\tl_if_empty:nTF { #1 }
{
  \__glossaries_tree_measure_symbol:Nn
  \l__glossaries_tmpa_dim { #2 }
}
{
  \__glossaries_tree_measure_child_symbol:Nn
  \l__glossaries_tmpa_dim { #2 }
}
\dim_compare:nNnT
{ \l__glossaries_tmpa_dim } >
{ \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
{
  \dim_set_eq:cN { l__glossaries_tree_symbolwidth #1 _dim }
  \l__glossaries_tmpa_dim
  \tl_set:cn { l__glossaries_tree_widest_symbol #1 _tl } { #2 }
  \tl_set:cn { l__glossaries_tree_symbolwidth #1 _tl }
  { \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
}
}

```

Update widest name and widest symbol if the combined width is greater than the current combined width.

```

\cs_new:Nn \glossaries_tree_update_widest_name_symbol:nnn
{
  \int_compare:nNnT
  { #1 } > { \l__glossaries_tree_max_known_level_int }
  {
    \int_set:Nn \l__glossaries_tree_max_known_level_int { #1 }
  }
  \exp_args:Ne \__glossaries_tree_update_widest_name_symbol:nnn
  { \romannumeral #1 } { #2 } { #3 }
}
\cs_generate_variant:Nn \glossaries_tree_update_widest_name_symbol:nnn
{ nee , nVV , nvv , noo }
\cs_new:Nn \__glossaries_tree_update_widest_name_symbol:nnn
{
  \dim_if_exist:cF
  { l__glossaries_tree_namesymonlywidth #1 _dim }
  {
    \dim_new:c { l__glossaries_tree_namesymonlywidth #1 _dim }
  }
  \tl_if_empty:nTF { #1 }
  {
    \__glossaries_tree_measure_name:Nn \l__glossaries_tmpa_dim { #2 }
    \__glossaries_tree_measure_symbol:Nn \l__glossaries_tmpb_dim { #3 }
  }
}

```

```

{
  \__glossaries_tree_measure_child_name:Nn
  \l__glossaries_tmpa_dim { #2 }
  \__glossaries_tree_measure_child_symbol:Nn
  \l__glossaries_tmpb_dim { #3 }
}
\dim_set:Nn \gls@tmplen
{ \l__glossaries_tmpa_dim + \l__glossaries_tmpb_dim }
\dim_compare:nNnT
{ \gls@tmplen }
>
{
  \dim_use:c { l__glossaries_tree_namesymonlywidth #1 _dim }
}
{
  \tl_set:cn { @glswidestname #1 } { #2 }
  \tl_set:cn { l__glossaries_tree_widest_symbol #1 _tl } { #3 }
  \dim_set_eq:cN { l__glossaries_tree_namesymonlywidth #1 _dim }
  { \gls@tmplen }
}

```

Unexpected results may occur with the following. Is name-width=widest also set?

```

\bool_lazy_and:nnTF
{
  \tl_if_exist_p:c { l__glossaries_tree_namewidth #1 _tl }
}
{ \tl_if_empty_p:c { l__glossaries_tree_namewidth #1 _tl } }
{
  \dim_if_exist:cF
  { l__glossaries_tree_namewidth #1 _dim }
  {
    \dim_new:c { l__glossaries_tree_namewidth #1 _dim }
  }
  \__glossaries_tree_init_namewidth:n { #1 }
  \dim_compare:nNnT
  { \l__glossaries_tmpa_dim } >
  { \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
  {
    \dim_set_eq:cN { l__glossaries_tree_namewidth #1 _dim }
    \l__glossaries_tmpa_dim
    \tl_set:cn { l__glossaries_tree_namewidth #1 _tl }
    { \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
  }
}
}

```

Is symbol-width=widest also set?

```

\bool_lazy_and:nnTF
{
  \tl_if_exist_p:c { l__glossaries_tree_symbolwidth #1 _tl }
}
{ \tl_if_empty_p:c { l__glossaries_tree_symbolwidth #1 _tl } }

```

```

{
  \dim_if_exist:cF
  { l__glossaries_tree_symbolwidth #1 _dim }
  {
    \dim_new:c { l__glossaries_tree_symbolwidth #1 _dim }
  }
  \__glossaries_tree_init_symbolwidth:n { #1 }
  \dim_compare:nNnT
  { \l__glossaries_tmpb_dim } >
  { \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
  {
    \dim_set_eq:cN { l__glossaries_tree_symbolwidth #1 _dim }
    \l__glossaries_tmpb_dim
    \tl_set:cn { l__glossaries_tree_symbolwidth #1 _tl }
    { \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
  }
}
}
}

```

Internal functions and variables. Header font:

```

\cs_new:Nn \__glossaries_tree_header_font:n { \textbf { #1 } }
\cs_new:Nn \__glossaries_tree_sub_header_font:n
{
  \__glossaries_tree_header_font:n { #1 }
}

```

Header alignment.

```

\cs_new:Nn \__glossaries_tree_header_fmt:n
{
  \raggedright
  #1
}
\cs_new:Nn \__glossaries_tree_sub_header_fmt:n
{
  \__glossaries_tree_header_fmt:n { #1 }
}
\cs_new:Nn \__glossaries_tree_hypernav_font:n
{
  \textbf { #1 }
}
\cs_new:Nn \__glossaries_tree_hypernav_fmt:n
{
  \raggedright
  #1
}

```

Name and symbol case-changing command (needed if measuring):

```

\cs_new:Nn \__glossaries_tree_name_case:n { #1 }
\cs_new:Nn \__glossaries_tree_symbol_case:n { #1 }

```



```

\cs_new:Nn \__glossaries_tree_child_name_case:n
{ \__glossaries_tree_name_case:n { #1 } }
\cs_new:Nn \__glossaries_tree_child_symbol_case:n
{ \__glossaries_tree_symbol_case:n { #1 } }

```

Name and symbol encapsulator (to add parentheses, if applicable):

```

\cs_new:Nn \__glossaries_tree_name_encap:n { #1 }
\cs_new:Nn \__glossaries_tree_child_name_encap:n { #1 }
\cs_new:Nn \__glossaries_tree_symbol_encap:n
{
  \glossaries_tree_paren:n { #1 }
}
\cs_new:Nn \__glossaries_tree_child_symbol_encap:n
{
  \glossaries_tree_paren:n { #1 }
}
\bool_new:N \l__glossaries_tree_child_style_inherited_bool
\bool_set_true:N \l__glossaries_tree_child_style_inherited_bool

```

Name font:

```

\cs_new:Nn \__glossaries_tree_name_font:n { \textbf { #1 } }
\cs_new:Nn \__glossaries_tree_child_name_font:n
{
  \__glossaries_tree_name_font:n { #1 }
}

```

Symbol font:

```

\cs_new:Nn \__glossaries_tree_symbol_font:n { #1 }
\cs_new:Nn \__glossaries_tree_child_symbol_font:n
{
  \__glossaries_tree_symbol_font:n { #1 }
}

```

Description font:

```

\cs_new:Nn \__glossaries_tree_desc_font:n { #1 }
\cs_new:Nn \__glossaries_tree_child_desc_font:n
{
  \__glossaries_tree_desc_font:n { #1 }
}

```

Location list font:

```

\cs_new:Nn \__glossaries_tree_loc_font:n { #1 }
\cs_new:Nn \__glossaries_tree_child_loc_font:n
{
  \__glossaries_tree_loc_font:n { #1 }
}

```

Display name (with glossaries-extra the category attributes may adjust the formatting or apply a post-name hook):

```

\cs_new:Nn \__glossaries_tree_gloss_name:n
{
  \glossentryname { #1 }
}

```

Display symbol:

```
\cs_new:Nn \__glossaries_tree_gloss_symbol:n
{
  \glossentrysymbol { #1 }
}
```

Display description:

```
\cs_new:Nn \__glossaries_tree_gloss_desc:n
{
  \glossentrydesc { #1 } \glspostdescription
}
```

Do content if description should be shown:

```
\cs_new:Nn \__glossaries_tree_show_desc:nn
{
  \glossaries_if_has_nonsuppressed_desc:nT
  { #1 } { #2 }
}
```

Do content if location should be shown (entry label, location list, content). The label argument is currently ignored.

```
\cs_new:Nn \__glossaries_tree_show_loc:nnn
{
  \tl_if_blank:nF { #2 } { #3 }
}
```

Display name for child entry:

```
\cs_new:Nn \__glossaries_tree_child_gloss_name:n
{
  \__glossaries_tree_gloss_name:n { #1 }
}
```

Display symbol for child entry:

```
\cs_new:Nn \__glossaries_tree_child_gloss_symbol:n
{
  \__glossaries_tree_gloss_symbol:n { #1 }
}
```

Display description for child entry:

```
\cs_new:Nn \__glossaries_tree_child_gloss_desc:n
{
  \__glossaries_tree_gloss_desc:n { #1 }
}
```

Do content if child description should be shown:

```
\cs_new:Nn \__glossaries_tree_show_child_desc:nn
{
  \__glossaries_tree_show_desc:nn { #1 } { #2 }
}
```

Do content if child location should be shown (entry label, location list, content):

```
\cs_new:Nn \__glossaries_tree_show_child_loc:nnn
{
  \tl_if_blank:nF { #2 } { #3 }
}
```

```

    \__glossaries_tree_show_loc:nnn { #1 } { #2 } { #3 }
  }

```

Separator between name and symbol:

```

\tl_new:N \l__glossaries_tree_name_sym_sep_tl
\tl_set:Nn \l__glossaries_tree_name_sym_sep_tl { ~ }
\tl_new:N \l__glossaries_tree_child_name_sym_sep_tl
\tl_set:Nn \l__glossaries_tree_child_name_sym_sep_tl
{
  \l__glossaries_tree_name_sym_sep_tl
}

```

Typeset name and/or symbol for top-level entry:

```

\cs_new:Nn \__glossaries_tree_name_sym_item:nnn
{
  \exp_args:NV \__glossaries_tree_name_sym_item:nnnn
  \l__glossaries_tree_name_sym_sep_tl
  { #1 } { #2 } { #3 }
}
\cs_new:Nn \__glossaries_tree_name_sym_item:nnnn
{
  \glstarget { #2 } { #3 }
  \tl_if_empty:nF { #4 }
  {
    #1 #4
  }
}

```

Typeset name and/or symbol for child entry:

```

\cs_new:Nn \__glossaries_tree_name_sym_subitem:nnn
{
  \exp_args:NV \__glossaries_tree_name_sym_subitem:nnnn
  \l__glossaries_tree_child_name_sym_sep_tl
  { #1 } { #2 } { #3 }
}
\cs_new:Nn \__glossaries_tree_name_sym_subitem:nnnn
{
  \__glossaries_tree_name_sym_item:nnnn
  { #1 } { #2 } { #3 } { #4 }
}

```

Separators between name/symbol and description:

```

\tl_new:N \l__glossaries_tree_post_namesymbol_tl
\tl_set:Nn \l__glossaries_tree_post_namesymbol_tl { ~ }
\tl_new:N \l__glossaries_tree_post_child_namesymbol_tl
\tl_set:Nn \l__glossaries_tree_post_child_namesymbol_tl
{
  \l__glossaries_tree_post_namesymbol_tl
}
\tl_new:N \l__glossaries_tree_pre_desc_tl
\tl_new:N \l__glossaries_tree_pre_child_desc_tl

```

```

\tl_set:Nn \l__glossaries_tree_pre_child_desc_tl
{
  \l__glossaries_tree_pre_desc_tl
}

```

Separator before location:

```

\tl_new:N \l__glossaries_tree_pre_loc_tl
\tl_set:Nn \l__glossaries_tree_pre_loc_tl { ~ }
\tl_new:N \l__glossaries_tree_pre_child_loc_tl
\tl_set:Nn \l__glossaries_tree_pre_child_loc_tl
{
  \l__glossaries_tree_pre_loc_tl
}

```

Separator after location:

```

\tl_new:N \l__glossaries_tree_post_loc_tl
\tl_new:N \l__glossaries_tree_post_child_loc_tl
\tl_set:Nn \l__glossaries_tree_post_child_loc_tl
{
  \l__glossaries_tree_post_loc_tl
}

```

Keep track of maximum known level.

```

\int_new:N \l__glossaries_tree_max_known_level_int

```

Sub-level offsets.

```

\dim_new:N \l__glossaries_tree_offset_dim
\tl_new:N \l__glossaries_tree_offset_tl

```

If width is less than or equal to 0pt then no fixed width box.

```

\dim_new:N \l__glossaries_tree_namewidth_dim
\tl_new:N \l__glossaries_tree_namewidth_tl
\tl_set:Nn \l__glossaries_tree_namewidth_tl { \c_zero_dim }
\tl_new:N \l__glossaries_tree_namealign_tl
\tl_set:Nn \l__glossaries_tree_namealign_tl { l }

```

Initialise the name and symbol widths for each known level.

```

\cs_new:Nn \__glossaries_tree_init_namesymbolwidths:
{
  \tl_if_empty:NTF \l__glossaries_tree_prenamesym_width_tl
  {
    \dim_zero:N \l__glossaries_tree_prenamesym_width_dim
  }
  {
    \dim_set:Nn \l__glossaries_tree_prenamesym_width_dim
    {
      \l__glossaries_tree_prenamesym_width_tl
    }
  }
  \tl_if_empty:NTF \l__glossaries_tree_presubnamesym_width_tl
  {
    \dim_zero:N \l__glossaries_tree_presubnamesym_width_dim
  }
}

```

```

}
{
  \dim_set:Nn \l__glossaries_tree_presubnamesym_width_dim
  {
    \l__glossaries_tree_presubnamesym_width_tl
  }
}
\tl_if_empty:NF \l__glossaries_tree_namesymbolwidth_padding_tl
{
  \dim_set:Nn
  \l__glossaries_tree_namesymbolwidth_padding_dim
  { \l__glossaries_tree_namesymbolwidth_padding_tl }
}
\tl_if_empty:NF \l__glossaries_tree_namewidth_padding_tl
{
  \dim_set:Nn \l__glossaries_tree_namewidth_padding_dim
  { \l__glossaries_tree_namewidth_padding_tl }
}
\tl_if_empty:NF \l__glossaries_tree_symbolwidth_padding_tl
{
  \dim_set:Nn \l__glossaries_tree_symbolwidth_padding_dim
  { \l__glossaries_tree_symbolwidth_padding_tl }
}
\int_step_inline:nnn
{ \c_zero_int } { \l__glossaries_tree_max_known_level_int }
{
  \exp_args:Ne \l__glossaries_tree_init_namewidth:n
  { \romannumeral ##1 }
  \exp_args:Ne \l__glossaries_tree_init_symbolwidth:n
  { \romannumeral ##1 }
  \exp_args:Ne \l__glossaries_tree_calc_namesymbolwidth:n
  { \romannumeral ##1 }
}
}

```

Initialise the name width for the given level (Roman numeral argument or empty for top-level).

```

\cs_new:Nn \l__glossaries_tree_init_namewidth:n
{
  \tl_if_exist:cTF { l__glossaries_tree_namewidth #1 _tl }
  {
    \dim_if_exist:cF
    { l__glossaries_tree_namewidth #1 _dim }
    {
      \dim_new:c { l__glossaries_tree_namewidth #1 _dim }
    }
  }
  \tl_if_empty:cTF { l__glossaries_tree_namewidth #1 _tl }
  {
    \bool_lazy_and:nnTF
    { \tl_if_exist_p:c { @glswidestname #1 } }
  }
}

```

```

    { \bool_not_p:n { \tl_if_empty_p:c { @glswidestname #1 } } }
  {
    \tl_if_empty:nTF { #1 }
    {
      \exp_args:NV \__glossaries_tree_measure_name:n
      \@glswidestname
    }
    {
      \exp_args:Nnv \__glossaries_tree_measure_child_name:nn { #1 }
      { @glswidestname #1 }
    }
  }
  {
    \dim_zero:c { l__glossaries_tree_namewidth #1 _dim }
  }
}
{
  \dim_set:cn { l__glossaries_tree_namewidth #1 _dim }
  { \tl_use:c { l__glossaries_tree_namewidth #1 _tl } }
}
}
{
  \dim_if_exist:cT
  { l__glossaries_tree_namewidth #1 _dim }
  {
    \dim_zero:c { l__glossaries_tree_namewidth #1 _dim }
  }
}
}

```

Measure top-level name:

```

\cs_new:Nn \__glossaries_tree_measure_name:n
{
  \__glossaries_tree_measure_name:Nn
  \l__glossaries_tree_namewidth_dim { #1 }
}
\cs_new:Nn \__glossaries_tree_measure_name:Nn
{
  \glsmeasurewidth #1
  {
    \__glossaries_tree_name_encap:n
    {
      \__glossaries_tree_name_font:n
      {
        \__glossaries_tree_name_case:n { #2 }
      }
    }
  }
}
}

```

Measure child name:

```

\cs_new:Nn \__glossaries_tree_measure_child_name:nn
{
  \exp_args:Nc \__glossaries_tree_measure_child_name:Nn
  { 1__glossaries_tree_namewidth #1 _dim }
  { #2 }
}
\cs_new:Nn \__glossaries_tree_measure_child_name:Nn
{
  \glsmeasurewidth #1
  {
    \__glossaries_tree_child_name_encap:n
    {
      \__glossaries_tree_child_name_font:n
      {
        \__glossaries_tree_child_name_case:n
        { #2 }
      }
    }
  }
}

```

Natural or fixed width box for name:

```

\cs_new:Nn \__glossaries_tree_namebox:n
{
  \__glossaries_tree_namebox:nn { } { #1 }
}
\cs_new:Nn \__glossaries_tree_namebox:nn
{
  \bool_lazy_and:nnTF
  {
    \dim_if_exist_p:c
    { 1__glossaries_tree_namewidth #1 _dim }
  }
  {
    \dim_compare_p:nNn
    { \dim_use:c { 1__glossaries_tree_namewidth #1 _dim } }
    >
    { \c_zero_dim }
  }
  {
    \tl_if_exist:cTF { 1__glossaries_tree_namealign #1 _tl }
    {
      \glossaries_tree_namebox:vvN
      { 1__glossaries_tree_namewidth #1 _dim }
      { 1__glossaries_tree_namealign #1 _tl }
      { #2 }
    }
    {
      \glossaries_tree_namebox:vVn
      { 1__glossaries_tree_namewidth #1 _dim }
    }
  }
}

```

```

        \l__glossaries_tree_namealign_tl
        { #2 }
    }
}
{
    \glossaries_tree_namenobox:n { #2 }
}
}

```

Similarly for symbol.

```

\dim_new:N \l__glossaries_tree_symbolwidth_dim
\tl_new:N \l__glossaries_tree_symbolwidth_tl
\tl_set:Nn \l__glossaries_tree_symbolwidth_tl { \c_zero_dim }
\tl_new:N \l__glossaries_tree_symbolalign_tl
\tl_set:Nn \l__glossaries_tree_symbolalign_tl { l }

```

Initialise symbol width for given level.

```

\cs_new:Nn \__glossaries_tree_init_symbolwidth:n
{
    \tl_if_exist:cTF { l__glossaries_tree_symbolwidth #1 _tl }
    {
        \dim_if_exist:cF
        { l__glossaries_tree_symbolwidth #1 _dim }
        {
            \dim_new:c { l__glossaries_tree_symbolwidth #1 _dim }
        }
    }
    \tl_if_empty:cTF { l__glossaries_tree_symbolwidth #1 _tl }
    {
        \bool_lazy_and:nnTF
        { \tl_if_exist_p:c { l__glossaries_tree_widest_symbol #1 _tl } }
        {
            \bool_not_p:n
            {
                \tl_if_empty_p:c { l__glossaries_tree_widest_symbol #1 _tl }
            }
        }
    }
    {
        \tl_if_empty:nTF { #1 }
        {
            \exp_args:NV \__glossaries_tree_measure_symbol:n
            \l__glossaries_tree_widest_symbol_tl
        }
        {
            \exp_args:Nnv \__glossaries_tree_measure_child_symbol:nn { #1 }
            { l__glossaries_tree_widest_symbol #1 _tl }
        }
    }
}
{
    \dim_zero:c { l__glossaries_tree_symbolwidth #1 _dim }
}
}

```



```

    {
      \dim_set:cn { l__glossaries_tree_symbolwidth #1 _dim }
      { \tl_use:c { l__glossaries_tree_symbolwidth #1 _tl } }
    }
  }
  {
    \dim_if_exist:cT
    { l__glossaries_tree_symbolwidth #1 _dim }
    {
      \dim_zero:c { l__glossaries_tree_symbolwidth #1 _dim }
    }
  }
}

```

Measure top-level symbol:

```

\cs_new:Nn \__glossaries_tree_measure_symbol:n
{
  \__glossaries_tree_measure_symbol:Nn
  \l__glossaries_tree_symbolwidth_dim { #1 }
}
\cs_new:Nn \__glossaries_tree_measure_symbol:Nn
{
  \glsmeasurewidth #1
  {
    \__glossaries_tree_symbol_encap:n
    {
      \__glossaries_tree_symbol_font:n
      {
        \__glossaries_tree_symbol_case:n { #2 }
      }
    }
  }
}

```

Measure child symbol:

```

\cs_new:Nn \__glossaries_tree_measure_child_symbol:nn
{
  \exp_args:Nc \__glossaries_tree_measure_child_symbol:Nn
  { l__glossaries_tree_symbolwidth #1 _dim }
  { #2 }
}
\cs_new:Nn \__glossaries_tree_measure_child_symbol:Nn
{
  \glsmeasurewidth #1
  {
    \__glossaries_tree_child_symbol_encap:n
    {
      \__glossaries_tree_child_symbol_font:n
      {
        \__glossaries_tree_child_symbol_case:n
        { #2 }
      }
    }
  }
}

```

```

    }
  }
}

```

Natural or fixed width box for symbol:

```

\cs_new:Nn \__glossaries_tree_symbolbox:n
{
  \__glossaries_tree_symbolbox:nn { } { #1 }
}
\cs_new:Nn \__glossaries_tree_symbolbox:nn
{
  \bool_lazy_and:nnTF
  {
    \dim_if_exist_p:c
    { l__glossaries_tree_symbolwidth #1 _dim }
  }
  {
    \dim_compare_p:nNn
    { \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
    >
    { \c_zero_dim }
  }
}
{
  \tl_if_exist:cTF { l__glossaries_tree_symbolalign #1 _tl }
  {
    \glossaries_tree_symbolbox:vvnn
    { l__glossaries_tree_symbolwidth #1 _dim }
    { l__glossaries_tree_symbolalign #1 _tl }
    { #2 }
  }
  {
    \glossaries_tree_symbolbox:vVnn
    { l__glossaries_tree_symbolwidth #1 _dim }
    \l__glossaries_tree_symbolalign_tl
    { #2 }
  }
}
{
  \glossaries_tree_symbolnobox:n { #2 }
}
}

```

Used for the entry item counter. Top-level:

```

\tl_new:N \l__glossaries_tree_prenamesym_align_tl
\tl_new:N \l__glossaries_tree_prenamesym_width_tl
\dim_new:N \l__glossaries_tree_prenamesym_width_dim
\cs_new:Nn \__glossaries_tree_prenamesym:n
{
  \tl_if_empty:NTF \l__glossaries_tree_prenamesym_width_tl
  {

```

```

\glossaries_tree_entryitem_nobox:n { #1 }
}
{
\glossaries_tree_entryitem_box:VVn
\l__glossaries_tree_prenamesym_width_dim
\l__glossaries_tree_prenamesym_align_tl
{ #1 }
}
}
\cs_new:Nn \__glossaries_tree_prenamesym_font:n { #1 }
Level 1:
\tl_new:N \l__glossaries_tree_presubnamesym_align_tl
\tl_new:N \l__glossaries_tree_presubnamesym_width_tl
\dim_new:N \l__glossaries_tree_presubnamesym_width_dim
\cs_new:Nn \__glossaries_tree_presubnamesym:n
{
\tl_if_empty:NTF \l__glossaries_tree_presubnamesym_width_tl
{
\glossaries_tree_subentryitem_nobox:n { #1 }
}
{
\glossaries_tree_subentryitem_box:VVn
\l__glossaries_tree_presubnamesym_width_dim
\l__glossaries_tree_presubnamesym_align_tl
{ #1 }
}
}
\cs_new:Nn \__glossaries_tree_presubnamesym_font:n { #1 }
\tl_new:N \l__glossaries_tree_namesymbolalign_tl
\tl_set:Nn \l__glossaries_tree_namesymbolalign_tl { l }
\cs_new:Nn \__glossaries_tree_namesymbox:n
{
\__glossaries_tree_namesymbox:nn { } { #1 }
}
\cs_new:Nn \__glossaries_tree_namesymbox:nn
{
\bool_lazy_and:nnTF
{
\dim_if_exist_p:c
{ l__glossaries_tree_namesymbolwidth #1 _dim }
}
{
\dim_compare_p:nNn
{ \dim_use:c { l__glossaries_tree_namesymbolwidth #1 _dim } }
>
{ \c_zero_dim }
}
}
{
\tl_if_exist:cTF { l__glossaries_tree_namesymbolalign #1 _tl }

```

```

{
  \glossaries_tree_namesymbolbox:vvn
  { l__glossaries_tree_namesymbolwidth #1 _dim }
  { l__glossaries_tree_namesymbolalign #1 _tl }
  { #2 }
}
{
  \glossaries_tree_namesymbolbox:vVn
  { l__glossaries_tree_namesymbolwidth #1 _dim }
  \l__glossaries_tree_namesymbolalign_tl
  { #2 }
}
}
{
  \glossaries_tree_namesymbolnobox:n { #2 }
}
}

```

Calculate the combined width of the name and symbol and separators.

```

\dim_new:N \l__glossaries_tree_namewidth_padding_dim
\tl_new:N \l__glossaries_tree_namewidth_padding_tl
\dim_new:N \l__glossaries_tree_symbolwidth_padding_dim
\tl_new:N \l__glossaries_tree_symbolwidth_padding_tl
\dim_new:N \l__glossaries_tree_namesymbolwidth_dim
\dim_new:N \l__glossaries_tree_namesymbolwidth_padding_dim
\tl_new:N \l__glossaries_tree_namesymbolwidth_padding_tl
\tl_new:N \l__glossaries_tree_namesymbolwidth_tl
\tl_set:Nn \l__glossaries_tree_namesymbolwidth_tl { \c_zero_dim }

```

Initialise the name and symbol width for the given level (Roman numeral argument or empty for top-level).

```

\cs_new:Nn \__glossaries_tree_calc_namesymbolwidth:n
{
  \tl_if_exist:cTF { l__glossaries_tree_namesymbolwidth #1 _tl }
  {
    \dim_if_exist:cF
    { l__glossaries_tree_namesymbolwidth #1 _dim }
    {
      \dim_new:c { l__glossaries_tree_namesymbolwidth #1 _dim }
    }
  }
  \tl_if_empty:cTF { l__glossaries_tree_namesymbolwidth #1 _tl }
  {

```

Calculate:

```

\dim_zero:c { l__glossaries_tree_namesymbolwidth #1 _dim }

```

Add the item counter padding, if set.

```

\bool_lazy_and:nnT
{
  \tl_if_exist_p:c
  { l__glossaries_tree_itemcounter_padding #1 _tl }

```

```

}
{
  \bool_not_p:n
  {
    \tl_if_empty_p:c
    { l__glossaries_tree_itemcounter_padding #1 _tl }
  }
}
{
  \dim_add:cn
  { l__glossaries_tree_namesymbolwidth #1 _dim }
  {
    \tl_use:c { l__glossaries_tree_itemcounter_padding #1 _tl }
  }
}
\gls@start@measuring

```

Measure the post name+symbol content and add.

```

\tl_if_empty:nTF { #1 }
{
  \settowidth \l__glossaries_tmpa_dim
  { \l__glossaries_tree_post_namesymbol_tl }
  \dim_add:Nn
  \l__glossaries_tree_namesymbolwidth_dim
  { \l__glossaries_tmpa_dim }
}
{
  \settowidth \l__glossaries_tmpa_dim
  { \l__glossaries_tree_post_child_namesymbol_tl }
  \dim_add:cn
  { l__glossaries_tree_namesymbolwidth #1 _dim }
  { \l__glossaries_tmpa_dim }
}

```

Check if combined update name and symbol was used.

```

\bool_lazy_and:nnTF
{
  \dim_if_exist_p:c { l__glossaries_tree_namesymonlywidth #1 _dim }
}
{
  \bool_not_p:n
  {
    \dim_compare_p:nNn
    { \dim_use:c { l__glossaries_tree_namesymonlywidth #1 _dim } }
    =
    { \c_zero_dim }
  }
}
{
  \dim_add:cn
  { l__glossaries_tree_namesymbolwidth #1 _dim }
}

```

```

{
  \dim_use:c { l__glossaries_tree_namesymbolwidth #1 _dim }
  + \l__glossaries_tree_namewidth_padding_dim
  + \l__glossaries_tree_symbolwidth_padding_dim
}
\tl_if_empty:nTF { #1 }
{
  \bool_lazy_or:nnF
  { \cs_if_eq_p:NN \__glossaries_tree_name_encap:n \use_none:n }
  { \cs_if_eq_p:NN \__glossaries_tree_symbol_encap:n \use_none:n }
  {
    \settowidth \l__glossaries_tmpa_dim
    { \l__glossaries_tree_name_sym_sep_tl }
    \dim_add:Nn \l__glossaries_tree_namesymbolwidth_dim
    { \l__glossaries_tmpa_dim }
  }
}
}
{
  \bool_lazy_or:nnF
  { \cs_if_eq_p:NN \__glossaries_tree_child_name_encap:n \use_none:n }
  { \cs_if_eq_p:NN \__glossaries_tree_child_symbol_encap:n \use_none:n }
  {
    \settowidth \l__glossaries_tmpa_dim
    { \l__glossaries_tree_child_name_sym_sep_tl }
    \dim_add:cn
    { l__glossaries_tree_namesymbolwidth #1 _dim }
    { \l__glossaries_tmpa_dim }
  }
}
}
{

```

Need to calculate the applicable widths.

```

\tl_if_empty:nTF { #1 }
{

```

Level 0

```

\cs_if_eq:NNF \__glossaries_tree_name_encap:n \use_none:n
{
  \dim_compare:nNnTF
  { \l__glossaries_tree_namewidth_dim } > { \c_zero_dim }
  {
    \dim_set:Nn \l__glossaries_tree_namesymbolwidth_dim
    { \l__glossaries_tree_namewidth_dim }
  }
  {
    \settowidth
    \l__glossaries_tree_namesymbolwidth_dim
    {
      \__glossaries_tree_name_encap:n
      {

```

```

        \_glossaries_tree_name_font:n
        {
            \exp_args:NV \_glossaries_tree_name_case:n
            \@glswidestname
        }
    }
}
}
\dim_add:Nn \l__glossaries_tree_namesymbolwidth_dim
{ \l__glossaries_tree_namewidth_padding_dim }
}
\cs_if_eq:NNF \_glossaries_tree_symbol_encap:n \use_none:n
{
    \settowidth \l__glossaries_tmpa_dim
    { \l__glossaries_tree_name_sym_sep_tl }
    \dim_add:Nn \l__glossaries_tree_namesymbolwidth_dim
    { \l__glossaries_tmpa_dim }
    \bool_lazy_or:nnTF
    {
        \bool_not_p:n
        { \tl_if_exist_p:N \l__glossaries_tree_widest_symbol_tl }
    }
    {
        \dim_compare_p:nNn
        { \l__glossaries_tree_symbolwidth_dim } > { \c_zero_dim }
    }
    {
        \dim_add:Nn \l__glossaries_tree_namesymbolwidth_dim
        { \l__glossaries_tree_symbolwidth_dim }
    }
    {
        \settowidth \l__glossaries_tmpa_dim
        {
            \_glossaries_tree_symbol_encap:n
            {
                \_glossaries_tree_symbol_font:n
                {
                    \exp_args:NV \_glossaries_tree_symbol_case:n
                    \l__glossaries_tree_widest_symbol_tl
                }
            }
        }
        \dim_add:Nn \l__glossaries_tree_namesymbolwidth_dim
        { \l__glossaries_tmpa_dim }
    }
    \dim_add:Nn \l__glossaries_tree_namesymbolwidth_dim
    { \l__glossaries_tree_symbolwidth_padding_dim }
}
}
{

```

Sub-levels.

```

\cs_if_eq:NNF \__glossaries_tree_child_name_encap:n \use_none:n
{
  \bool_lazy_and:nnTF
  {
    \dim_if_exist_p:c { l__glossaries_tree_namewidth #1 _dim }
  }
  {
    \dim_compare_p:nNn
    { \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
    > { \c_zero_dim }
  }
  {
    \dim_set:cn
    { l__glossaries_tree_namesymbolwidth #1 _dim }
    { \dim_use:c { l__glossaries_tree_namewidth #1 _dim } }
  }
  {
    \tl_if_exist:cT
    { @glswidestname #1 }
    {
      \exp_args:Nc \settowidth
      { l__glossaries_tree_namesymbolwidth #1 _dim }
      {
        \__glossaries_tree_child_name_encap:n
        {
          \__glossaries_tree_child_name_font:n
          {
            \exp_args:Nv \__glossaries_tree_child_name_case:n
            { @glswidestname #1 }
          }
        }
      }
    }
  }
}
\dim_add:cn
{ l__glossaries_tree_namesymbolwidth #1 _dim }
{ \l__glossaries_tree_namewidth_padding_dim }
}
\cs_if_eq:NNF \__glossaries_tree_child_symbol_encap:n \use_none:n
{
  \settowidth \l__glossaries_tmpa_dim
  { \l__glossaries_tree_child_name_sym_sep_tl }
  \dim_add:cn
  { l__glossaries_tree_namesymbolwidth #1 _dim }
  { \l__glossaries_tmpa_dim }
  \bool_lazy_and:nnTF
  {
    \dim_if_exist_p:c { l__glossaries_tree_symbolwidth #1 _dim }
  }
}

```



```

    }
    {
      \dim_compare_p:nNn
      { \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
      > { \c_zero_dim }
    }
    {
      \dim_add:cn
      { l__glossaries_tree_namesymbolwidth #1 _dim }
      { \dim_use:c { l__glossaries_tree_symbolwidth #1 _dim } }
    }
    {
      \tl_if_exist:cT
      { l__glossaries_tree_widest_symbol #1 _tl }
      {
        \settowidth \l__glossaries_tmpa_dim
        {
          \__glossaries_tree_child_symbol_encap:n
          {
            \__glossaries_tree_child_symbol_font:n
            {
              \exp_args:Nv \__glossaries_tree_child_symbol_case:n
              { l__glossaries_tree_widest_symbol #1 _tl }
            }
          }
        }
        \dim_add:cn
        { l__glossaries_tree_namesymbolwidth #1 _dim }
        { \l__glossaries_tmpa_dim }
      }
    }
    \dim_add:cn
    { l__glossaries_tree_namesymbolwidth #1 _dim }
    { \l__glossaries_tree_symbolwidth_padding_dim }
  }
}
}
\gls@stop@measuring
}
{
  \dim_set:cn { l__glossaries_tree_namesymbolwidth #1 _dim }
  { \tl_use:c { l__glossaries_tree_namesymbolwidth #1 _tl } }
}
}
{
  \dim_if_exist:cT
  { l__glossaries_tree_namesymbolwidth #1 _dim }
  {
    \dim_zero:c { l__glossaries_tree_namesymbolwidth #1 _dim }
  }
}

```

```

    }
}

```

Hanging indent. Empty if `\hangindent` shouldn't be set.

```
\tl_new:N \l__glossaries_tree_hangindent_tl
```

Define options:

```

\keys_define:nn { glossaries / styles / tree * }
{
  group-headings .bool_set:N = \l__glossaries_tree_group_headings_bool ,
  group-headings .initial:n = { false } ,
  bookmark-groups .choice: ,
  bookmark-groups / true .code:n =
  {
    \cs_set_eq:NN
      \__glossaries_tree_pregrouphead:nnn
      \__glossaries_tree_bookmark_group:nnn
  } ,
  bookmark-groups / false .code:n =
  {
    \cs_new:Nn \__glossaries_tree_pregrouphead:nnn { }
  } ,
  bookmark-groups .default:n = true ,

```

Locally switch off headings and group skip at the start of the glossary:

```

hide-groups .bool_set:N = \l__glossaries_tree_hide_groups_bool ,
hide-groups .initial:n = { false } ,

```

Not using skip variable to allow for font-dependent units. NB the group skip comes before the group header (if shown).

```

group-skip .tl_set:N = \l__glossaries_tree_group_skip_tl ,
group-skip .initial:n = { 1ex \@plus 2\p@ \@minus \p@ } ,
post-group-heading-skip .tl_set:N = \l__glossaries_tree_postgrouphead_skip_tl ,
post-group-heading-skip .initial:n = { \p@ \@plus 2\p@ } ,

```

If empty, don't change paragraph indent.

```

par-indent .tl_set:N = \l__glossaries_tree_parindent_tl ,
par-indent .default:n = { } ,
par-skip .tl_set:N = \l__glossaries_tree_parskip_tl ,
par-skip .initial:n = { \c_zero_dim \@plus 0.3\p@ } ,

```

If empty, don't change hanging indent.

```

hang-indent .code:n =
{
  \tl_if_eq:nnTF { #1 } { calculated }
  {
    \tl_set:Nn \l__glossaries_tree_hangindent_tl
    {
      \l__glossaries_tree_namesymbolwidth_dim
      + \l__glossaries_tree_namesymbolwidth_padding_dim
      + \l__glossaries_tree_prenamesym_width_dim
      + \l__glossaries_tree_presubnamesym_width_dim
    }
  }
}

```

```

    }
  }
  {
    \tl_set:Nn \l__glossaries_tree_hangindent_tl { #1 }
  }
},
hang-indent .default:n = { calculated } ,

```

Any horizontal spacing taken up by a frame etc in the box used for the name, symbol and separators that needs to be taking into account when calculating the hang-indent. The dimension isn't set at this point to allow for changes to variables such as `\fboxsep`.

```

name-symbol-padding .tl_set:N =
  \l__glossaries_tree_namesymbolwidth_padding_tl ,

```

Any horizontal spacing taken up by a frame etc in the box used for the name that needs to be taking into account when calculating the name-symbol-width.

```

name-padding .tl_set:N =
  \l__glossaries_tree_namewidth_padding_tl ,

```

Any horizontal spacing taken up by a frame etc in the box used for the symbol that needs to be taking into account when calculating the name-symbol-width.

```

symbol-padding .tl_set:N =
  \l__glossaries_tree_symbolwidth_padding_tl ,

```

sub-level indent offset:

```

child-offset .tl_set:N = \l__glossaries_tree_offset_tl ,
child-offset .initial:n = { 1em } ,

```

The final command in the list may be a text-block command:

```

header-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \__glossaries_tree_header_font:n { ##1 }
  }
  {
    \cs_set:Nn \__glossaries_tree_header_font:n
    {
      #1 { ##1 }
    }
  }
}
} ,
header-align .choice: ,
header-align / left .code:n =
{
  \cs_set:Nn \__glossaries_tree_header_fmt:n
  {
    \raggedright ##1
  }
}
} ,

```

```

header-align / right .code:n =
{
  \cs_set:Nn \__glossaries_tree_header_fmt:n
  {
    \raggedleft ##1
  }
},
header-align / center .code:n =
{
  \cs_set:Nn \__glossaries_tree_header_fmt:n
  {
    \centering ##1
  }
},
header-align / indent .code:n =
{
  \cs_set:Nn \__glossaries_tree_header_fmt:n
  {
    \l__glossaries_tree_par_setup_tl
    \par
    ##1
  }
},

```

Provide support for “unsrt” sub-groups:

```

sub-header-font .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \cs_set:Nn \__glossaries_tree_sub_header_font:n
    {
      \__glossaries_tree_header_font:n { ##1 }
    }
  }
  {
    \tl_if_empty:nTF { #1 }
    {
      \cs_set:Nn \__glossaries_tree_sub_header_font:n { ##1 }
    }
    {
      \cs_set:Nn \__glossaries_tree_sub_header_font:n
      {
        #1 { ##1 }
      }
    }
  }
},
sub-header-align .choice: ,
sub-header-align / inherit .code:n =
{

```

```

\cs_set:Nn \__glossaries_tree_sub_header_fmt:n
{
  \__glossaries_tree_header_fmt:n { ##1 }
}
},
sub-header-align / left .code:n =
{
  \cs_set:Nn \__glossaries_tree_sub_header_fmt:n
  {
    \raggedright ##1
  }
},
sub-header-align / right .code:n =
{
  \cs_set:Nn \__glossaries_tree_sub_header_fmt:n
  {
    \raggedleft ##1
  }
},
sub-header-align / center .code:n =
{
  \cs_set:Nn \__glossaries_tree_sub_header_fmt:n
  {
    \centering ##1
  }
},
sub-header-align / indent .code:n =
{
  \cs_set:Nn \__glossaries_tree_sub_header_fmt:n
  {
    \l__glossaries_tree_par_setup_tl
    \par
    ##1
  }
},

```

Hyper-navigation strip font and alignment

```

hyper-nav .bool_set:N = \l__glossaries_tree_hypernav_bool ,
hyper-nav .initial:n = { false } ,
hyper-nav-skip .tl_set:N = \l__glossaries_tree_hypernav_skip_tl ,
hyper-nav-skip .initial:n = { \l__glossaries_tree_group_skip_tl } ,
hyper-nav-font .code:n =
{
  \cs_set:Nn \__glossaries_tree_hypernav_font:n
  {
    #1 { ##1 }
  }
},
hyper-nav-align .choice: ,
hyper-nav-align / left .code:n =

```

```

{
  \cs_set:Nn \__glossaries_tree_hypernav_fmt:n
  {
    \raggedright ##1
  }
} ,
hyper-nav-align / right .code:n =
{
  \cs_set:Nn \__glossaries_tree_hypernav_fmt:n
  {
    \raggedleft ##1
  }
} ,
hyper-nav-align / center .code:n =
{
  \cs_set:Nn \__glossaries_tree_hypernav_fmt:n
  {
    \centering ##1
  }
} ,
hyper-nav-align / indent .code:n =
{
  \cs_set:Nn \__glossaries_tree_hypernav_fmt:n
  {
    \l__glossaries_tree_par_setup_tl
    \par
    ##1
  }
} ,

```

Allow case-change for the name. NB the normal option may still implement a case-change via glossaries-extra category attributes.

```

name-case .choice: ,
name-case / normal .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_name:n
  {
    \glossentryname { ##1 }
  }
  \cs_set_eq:NN \__glossaries_tree_name_case:n \use:n
} ,
name-case / firstuc .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_name:n
  {
    \Glossentryname { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_name_case:n
  { \glssentencecase { ##1 } }
} ,

```

```

name-case / uc .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_name:n
  {
    \GLOSSEntryname { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_name_case:n
  { \gluppercase { ##1 } }
} ,
name-case / title .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_name:n
  {
    \GlossEntryName { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_name_case:n
  { \glscapitalisewords { ##1 } }
} ,

```

Similarly for the child name.

```

child-name-case .choice: ,
child-name-case / inherit .code =
{
  \cs_set:Nn \__glossaries_tree_child_gloss_name:n
  {
    \__glossaries_tree_gloss_name:n { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_child_name_case:n
  { \__glossaries_tree_name_case:n { ##1 } }
} ,
child-name-case / normal .code =
{
  \cs_set:Nn \__glossaries_tree_child_gloss_name:n
  {
    \glossentryname { ##1 }
  }
  \cs_set_eq:NN
  \__glossaries_tree_child_name_case:n \use:n
} ,
child-name-case / firstuc .code =
{
  \cs_set:Nn \__glossaries_tree_child_gloss_name:n
  {
    \Glossentryname { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_child_name_case:n
  { \glssentencecase { ##1 } }
} ,
child-name-case / uc .code =
{

```

```

\cs_set:Nn \__glossaries_tree_child_gloss_name:n
{
  \GLOSSentryname { ##1 }
}
\cs_set:Nn \__glossaries_tree_child_name_case:n
{ \gluppercase { ##1 } }
} ,
child-name-case / title .code =
{
  \cs_set:Nn \__glossaries_tree_child_gloss_name:n
  {
    \GlossEntryName { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_child_name_case:n
  { \glscapitalisewords { ##1 } }
} ,

```

Case-changing is usually not applicable with symbols, but provide for completeness.

```

symbol-case .choice: ,
symbol-case / normal .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_symbol:n
  {
    \glossentrysymbol { ##1 }
  }
  \cs_set_eq:NN \__glossaries_tree_symbol_case:n \use:n
} ,
symbol-case / firstuc .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_symbol:n
  {
    \Glossentrysymbol { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_symbol_case:n
  { \glssentencecase { ##1 } }
} ,
symbol-case / uc .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_symbol:n
  {
    \GLOSSentrysymbol { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_symbol_case:n
  { \gluppercase { ##1 } }
} ,
symbol-case / title .code =
{
  \cs_set:Nn \__glossaries_tree_gloss_symbol:n
  {

```



```

        \GlossEntrySymbol { ##1 }
    }
    \cs_set:Nn \__glossaries_tree_symbol_case:n
    { \glscapitalisewords { ##1 } }
} ,

```

Similarly for the child symbol.

```

child-symbol-case .choice: ,
child-symbol-case / inherit .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_symbol:n
    {
        \__glossaries_tree_gloss_symbol:n { ##1 }
    }
    \cs_set:Nn \__glossaries_tree_child_symbol_case:n
    { \__glossaries_tree_symbol_case:n { ##1 } }
} ,
child-symbol-case / normal .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_symbol:n
    {
        \glossentrysymbol { ##1 }
    }
    \cs_set_eq:NN \__glossaries_tree_child_symbol_case:n \use:n
} ,
child-symbol-case / firstuc .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_symbol:n
    {
        \Glossentrysymbol { ##1 }
    }
    \cs_set:Nn \__glossaries_tree_child_symbol_case:n
    { \glssentencecase { ##1 } }
} ,
child-symbol-case / uc .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_symbol:n
    {
        \GLOSSentrysymbol { ##1 }
    }
    \cs_set:Nn \__glossaries_tree_child_symbol_case:n
    { \glssuppercase { ##1 } }
} ,
child-symbol-case / title .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_symbol:n
    {
        \GlossEntrySymbol { ##1 }
    }
    \cs_set:Nn \__glossaries_tree_child_symbol_case:n

```

```

        { \glscapitalisewords { ##1 } }
    } ,

```

Change case for description:

```

description-case .choice: ,
description-case / normal .code =
{
    \cs_set:Nn \__glossaries_tree_gloss_desc:n
    {
        \glossentrydesc { ##1 } \glspostdescription
    }
} ,
description-case / firstuc .code =
{
    \cs_set:Nn \__glossaries_tree_gloss_desc:n
    {
        \Glossentrydesc { ##1 } \glspostdescription
    }
} ,
description-case / uc .code =
{
    \cs_set:Nn \__glossaries_tree_gloss_desc:n
    {
        \GLOSSentrydesc { ##1 } \glspostdescription
    }
} ,
description-case / title .code =
{
    \cs_set:Nn \__glossaries_tree_gloss_desc:n
    {
        \GlossEntryDesc { ##1 } \glspostdescription
    }
} ,
child-description-case .choice: ,
child-description-case / inherit .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_desc:n
    {
        \__glossaries_tree_gloss_desc:n { ##1 }
    }
} ,
child-description-case / normal .code =
{
    \cs_set:Nn \__glossaries_tree_child_gloss_desc:n
    {
        \glossentrydesc { ##1 } \glspostdescription
    }
} ,
child-description-case / firstuc .code =
{

```

```

\cs_set:Nn \__glossaries_tree_child_gloss_desc:n
{
  \Glossentrydesc { ##1 } \glspostdescription
}
} ,
child-description-case / uc .code =
{
  \cs_set:Nn \__glossaries_tree_child_gloss_desc:n
  {
    \GLOSSentrydesc { ##1 } \glspostdescription
  }
} ,
child-description-case / title .code =
{
  \cs_set:Nn \__glossaries_tree_child_gloss_desc:n
  {
    \GlossEntryDesc { ##1 } \glspostdescription
  }
} ,

```

Allow the description to be omitted.

```

omit-description .choice: ,
omit-description / true .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_show_desc:nn \use_none:nn
} ,
omit-description / false .code:n =
{
  \cs_set:Nn \__glossaries_tree_show_desc:nn
  {
    \glossaries_if_has_nonsuppressed_desc:nT
    { ##1 } { ##2 }
  }
} ,
omit-description .default:n = true ,
omit-child-description .choice: ,
omit-child-description / true .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_show_child_desc:nn \use_none:nn
} ,
omit-child-description / false .code:n =
{
  \cs_set:Nn \__glossaries_tree_show_child_desc:nn
  {
    \glossaries_if_has_nonsuppressed_desc:nT
    { ##1 } { ##2 }
  }
} ,
omit-child-description / inherit .code:n =
{

```

```

\cs_set:Nn \__glossaries_tree_show_child_desc:nn
{
  \__glossaries_tree_show_desc:nn { ##1 } { ##2 }
}
} ,
omit-child-description .default:n = true ,
Allow the location to be omitted (NB not the same as nonnumberlist).
omit-location .choice: ,
omit-location / true .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_show_loc:nnn \use_none:nnn
} ,
omit-location / false .code:n =
{
  \cs_set:Nn \__glossaries_tree_show_loc:nnn
  {
    \tl_if_blank:nF { ##2 } { ##3 }
  }
} ,
omit-location .default:n = true ,
omit-child-location .choice: ,
omit-child-location / true .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_show_child_loc:nnn \use_none:nnn
} ,
omit-child-location / false .code:n =
{
  \cs_set:Nn \__glossaries_tree_show_child_loc:nnn
  {
    \tl_if_blank:nF { ##2 } { ##3 }
  }
} ,
omit-child-location / inherit .code:n =
{
  \cs_set:Nn \__glossaries_tree_show_child_loc:nnn
  {
    \__glossaries_tree_show_loc:nnn { ##1 } { ##2 } { ##3 }
  }
} ,
omit-child-location .default:n = true ,

```

Name and symbol style.

```

name-style .choice: ,
name-style / name ~ ( symbol ) .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_name_encap:n \use:n
  \cs_set:Nn \__glossaries_tree_symbol_encap:n
  {
    \glossaries_tree_paren:n { ##1 }
  }
}

```

```

\bool_if:NT \l__glossaries_tree_child_style_inherited_bool
{
  \cs_set_eq:NN
    \__glossaries_tree_child_name_encap:n
    \__glossaries_tree_name_encap:n
  \cs_set_eq:NN
    \__glossaries_tree_child_symbol_encap:n
    \__glossaries_tree_symbol_encap:n
}
\cs_set:Nn \__glossaries_tree_name_sym_item:nnnn
{
  \glstarget { ##2 } { ##3 }
  \tl_if_empty:nF { ##4 }
  {
    ##1 ##4
  }
}
} ,
name-style / name ~ symbol .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_name_encap:n \use:n
  \cs_set_eq:NN \__glossaries_tree_symbol_encap:n \use:n
  \bool_if:NT \l__glossaries_tree_child_style_inherited_bool
  {
    \cs_set_eq:NN
      \__glossaries_tree_child_name_encap:n
      \__glossaries_tree_name_encap:n
    \cs_set_eq:NN
      \__glossaries_tree_child_symbol_encap:n
      \__glossaries_tree_symbol_encap:n
  }
  \cs_set:Nn \__glossaries_tree_name_sym_item:nnnn
  {
    \glstarget { ##2 } { ##3 }
    \tl_if_empty:nF { ##4 }
    {
      ##1 ##4
    }
  }
} ,
name-style / symbol ~ name .code:n =
{
  \cs_set_eq:NN \__glossaries_tree_name_encap:n \use:n
  \cs_set_eq:NN \__glossaries_tree_symbol_encap:n \use:n
  \bool_if:NT \l__glossaries_tree_child_style_inherited_bool
  {
    \cs_set_eq:NN
      \__glossaries_tree_child_name_encap:n
      \__glossaries_tree_name_encap:n
    \cs_set_eq:NN

```

```

        \_glossaries_tree_child_symbol_encap:n
        \_glossaries_tree_symbol_encap:n
    }
\cs_set:Nn \_glossaries_tree_name_sym_item:nnnn
{
    \tl_if_empty:nF { ##4 }
    {
        ##4 ##1
    }
    \glstarget { ##2 } { ##3 }
}
} ,
name-style / symbol ~ ( name ) .code:n =
{
    \cs_set:Nn \_glossaries_tree_name_encap:n
    {
        \glossaries_tree_paren:n { ##1 }
    }
    \cs_set_eq:NN \_glossaries_tree_symbol_encap:n \use:n
    \bool_if:NT \l_glossaries_tree_child_style_inherited_bool
    {
        \cs_set_eq:NN
            \_glossaries_tree_child_name_encap:n
            \_glossaries_tree_name_encap:n
        \cs_set_eq:NN
            \_glossaries_tree_child_symbol_encap:n
            \_glossaries_tree_symbol_encap:n
    }
    \cs_set:Nn \_glossaries_tree_name_sym_item:nnnn
    {
        \tl_if_empty:nTF { ##4 }
        {
            \glstarget { ##2 } { ##3 }
        }
        {
            ##4 ##1
            \glstarget { ##2 } { ##3 }
        }
    }
} ,
name-style / symbol .code:n =
{
    \cs_set_eq:NN \_glossaries_tree_name_encap:n \use_none:n
    \cs_set_eq:NN \_glossaries_tree_symbol_encap:n \use:n
    \bool_if:NT \l_glossaries_tree_child_style_inherited_bool
    {
        \cs_set_eq:NN
            \_glossaries_tree_child_name_encap:n
            \_glossaries_tree_name_encap:n
        \cs_set_eq:NN

```

```

        \_glossaries_tree_child_symbol_encap:n
        \_glossaries_tree_symbol_encap:n
    }
\cs_set:Nn \_glossaries_tree_name_sym_item:nnnn
{
    \glstarget { ##2 } { ##3 }
}
} ,
name-style / name .code:n =
{
    \cs_set_eq:NN \_glossaries_tree_name_encap:n \use:n
    \cs_set_eq:NN \_glossaries_tree_symbol_encap:n \use_none:n
    \bool_if:NT \l_glossaries_tree_child_style_inherited_bool
    {
        \cs_set_eq:NN
            \_glossaries_tree_child_name_encap:n
            \_glossaries_tree_name_encap:n
        \cs_set_eq:NN
            \_glossaries_tree_child_symbol_encap:n
            \_glossaries_tree_symbol_encap:n
    }
\cs_set:Nn \_glossaries_tree_name_sym_item:nnnn
{
    \glstarget { ##2 } { ##3 }
}
} ,

```

Name/symbol style for child entry:

```

child-name-style .choice: ,
child-name-style / inherit .code:n =
{
    \bool_set_true:N \l_glossaries_tree_child_style_inherited_bool
    \cs_set_eq:NN
        \_glossaries_tree_child_name_encap:n
        \_glossaries_tree_name_encap:n
    \cs_set_eq:NN
        \_glossaries_tree_child_symbol_encap:n
        \_glossaries_tree_symbol_encap:n
    \cs_set:Nn \_glossaries_tree_name_sym_subitem:nnnn
    {
        \_glossaries_tree_name_sym_item:nnnn
        { ##1 } { ##2 } { ##3 } { ##4 }
    }
} ,
child-name-style / omit .code:n =
{
    \bool_set_false:N \l_glossaries_tree_child_style_inherited_bool
    \cs_set_eq:NN \_glossaries_tree_child_name_encap:n \use_none:n
    \cs_set_eq:NN \_glossaries_tree_child_symbol_encap:n \use_none:n
    \cs_set:Nn \_glossaries_tree_name_sym_subitem:nnnn

```

```

        {
            \glstarget { ##2 } { \strut }
        }
    } ,
child-name-style / name ~ ( symbol ) .code:n =
{
    \bool_set_false:N \l__glossaries_tree_child_style_inherited_bool
    \cs_set_eq:NN \__glossaries_tree_child_name_encap:n \use:n
    \cs_set:Nn \__glossaries_tree_child_symbol_encap:n
    {
        \glossaries_tree_paren:n { ##1 }
    }
    \cs_set:Nn \__glossaries_tree_name_sym_subitem:nnnn
    {
        \glstarget { ##2 } { ##3 }
        \tl_if_empty:nF { ##4 }
        {
            ##1 ##4
        }
    }
} ,
child-name-style / name ~ symbol .code:n =
{
    \bool_set_false:N \l__glossaries_tree_child_style_inherited_bool
    \cs_set_eq:NN \__glossaries_tree_child_name_encap:n \use:n
    \cs_set_eq:NN \__glossaries_tree_child_symbol_encap:n \use:n
    \cs_set:Nn \__glossaries_tree_name_sym_subitem:nnnn
    {
        \glstarget { ##2 } { ##3 }
        \tl_if_empty:nF { ##4 }
        {
            ##1 ##4
        }
    }
} ,
child-name-style / symbol ~ name .code:n =
{
    \bool_set_false:N \l__glossaries_tree_child_style_inherited_bool
    \cs_set_eq:NN \__glossaries_tree_child_name_encap:n \use:n
    \cs_set_eq:NN \__glossaries_tree_child_symbol_encap:n \use:n
    \cs_set:Nn \__glossaries_tree_name_sym_subitem:nnnn
    {
        \tl_if_empty:nF { ##4 }
        {
            ##4 ##1
        }
        \glstarget { ##2 } { ##3 }
    }
} ,
child-name-style / symbol ~ ( name ) .code:n =

```



```

{
  \bool_set_false:N \l__glossaries_tree_child_style_inherited_bool
  \cs_set_eq:NN \__glossaries_tree_child_symbol_encap:n \use:n
  \cs_set:Nn \__glossaries_tree_child_name_encap:n
  {
    \glossaries_tree_paren:n { ##1 }
  }
  \cs_set:Nn \__glossaries_tree_name_sym_subitem:nnnn
  {
    \tl_if_empty:nTF { ##4 }
    { \glstarget { ##2 } { ##3 } }
    {
      ##4 ##1
      \glstarget { ##2 } { ##3 }
    }
  }
} ,
child-name-style / symbol .code:n =
{
  \bool_set_false:N \l__glossaries_tree_child_style_inherited_bool
  \cs_set_eq:NN \__glossaries_tree_child_symbol_encap:n \use:n
  \cs_set_eq:NN \__glossaries_tree_child_name_encap:n \use_none:n
  \cs_set:Nn \__glossaries_tree_name_sym_subitem:nnnn
  {
    \glstarget { ##2 } { ##4 }
  }
} ,
child-name-style / name .code:n =
{
  \bool_set_false:N \l__glossaries_tree_child_style_inherited_bool
  \cs_set_eq:NN \__glossaries_tree_child_symbol_encap:n \use_none:n
  \cs_set_eq:NN \__glossaries_tree_child_name_encap:n \use:n
  \cs_set:Nn \__glossaries_tree_name_sym_subitem:nnnn
  {
    \glstarget { ##2 } { ##3 }
  }
} ,

```

Name font:

```

name-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \__glossaries_tree_name_font:n { ##1 }
  }
  {
    \cs_set:Nn \__glossaries_tree_name_font:n
    {
      #1 { ##1 }
    }
  }
}

```

```

    }
  } ,
  child-name-font .code:n =
  {
    \tl_if_eq:nnTF { #1 } { inherit }
    {
      \cs_set:Nn \__glossaries_tree_child_name_font:n
      {
        \__glossaries_tree_name_font:n { ##1 }
      }
    }
    {
      \tl_if_empty:nTF { #1 }
      {
        \cs_set:Nn \__glossaries_tree_child_name_font:n { ##1 }
      }
      {
        \cs_set:Nn \__glossaries_tree_child_name_font:n
        {
          #1 { ##1 }
        }
      }
    }
  }
} ,

```

Symbol font:

```

symbol-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \__glossaries_tree_symbol_font:n { ##1 }
  }
  {
    \cs_set:Nn \__glossaries_tree_symbol_font:n
    {
      #1 { ##1 }
    }
  }
} ,
child-symbol-font .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \cs_set:Nn \__glossaries_tree_child_symbol_font:n
    {
      \__glossaries_tree_symbol_font:n { ##1 }
    }
  }
  {
    \tl_if_empty:nTF { #1 }

```

```

    {
      \cs_set:Nn \__glossaries_tree_child_symbol_font:n { ##1 }
    }
    {
      \cs_set:Nn \__glossaries_tree_child_symbol_font:n
      {
        #1 { ##1 }
      }
    }
  }
} ,

```

Description font:

```

description-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \__glossaries_tree_desc_font:n { ##1 }
  }
  {
    \cs_set:Nn \__glossaries_tree_desc_font:n
    {
      #1 { ##1 }
    }
  }
} ,
child-description-font .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \cs_set:Nn \__glossaries_tree_child_desc_font:n
    {
      \__glossaries_tree_desc_font:n { ##1 }
    }
  }
  {
    \tl_if_empty:nTF { #1 }
    {
      \cs_set:Nn \__glossaries_tree_child_desc_font:n { ##1 }
    }
    {
      \cs_set:Nn \__glossaries_tree_child_desc_font:n
      {
        #1 { ##1 }
      }
    }
  }
} ,

```

Location list font:

```

location-font .code:n =

```

```

{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set:Nn \__glossaries_tree_loc_font:n { ##1 }
  }
  {
    \cs_set:Nn \__glossaries_tree_loc_font:n
    {
      #1 { ##1 }
    }
  }
} ,
child-location-font .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \cs_set:Nn \__glossaries_tree_child_loc_font:n
    {
      \__glossaries_tree_loc_font:n { ##1 }
    }
  }
  {
    \tl_if_empty:nTF { #1 }
    {
      \cs_set:Nn \__glossaries_tree_child_loc_font:n { ##1 }
    }
    {
      \cs_set:Nn \__glossaries_tree_child_loc_font:n
      {
        #1 { ##1 }
      }
    }
  }
} ,

```

Separators between name, symbol and description. Separator between the name and symbol:

```

name-symbol-sep .tl_set:N = \l__glossaries_tree_name_sym_sep_tl ,
child-name-symbol-sep .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \tl_set:Nn \l__glossaries_tree_child_name_sym_sep_tl
    {
      \l__glossaries_tree_name_sym_sep_tl
    }
  }
  {
    \tl_set:Nn \l__glossaries_tree_child_name_sym_sep_tl { #1 }
  }
}

```

```
},
```

The post-name-symbol token list is placed inside the name-symbol box.

```
post-name-symbol .tl_set:N = \l__glossaries_tree_post_namesymbol_tl ,
post-child-name-symbol .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \tl_set:Nn \l__glossaries_tree_post_child_namesymbol_tl
    {
      \l__glossaries_tree_post_namesymbol_tl
    }
  }
  {
    \tl_set:Nn \l__glossaries_tree_post_namesymbol_tl { #1 }
  }
},
```

The separator before description is placed outside the name-symbol box.

```
pre-description .tl_set:N = \l__glossaries_tree_pre_desc_tl ,
pre-child-description .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \tl_set:Nn \l__glossaries_tree_pre_child_desc_tl
    {
      \l__glossaries_tree_pre_desc_tl
    }
  }
  {
    \tl_set:Nn \l__glossaries_tree_pre_child_desc_tl { #1 }
  }
},
```

Separator before location:

```
pre-location .tl_set:N = \l__glossaries_tree_pre_loc_tl ,
pre-child-location .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \tl_set:Nn \l__glossaries_tree_pre_child_loc_tl
    {
      \l__glossaries_tree_pre_loc_tl
    }
  }
  {
    \tl_set:Nn \l__glossaries_tree_pre_child_loc_tl { #1 }
  }
},
```

Separator after location:

```

post-location .tl_set:N = \l__glossaries_tree_post_loc_tl ,
post-child-location .code:n =
{
  \tl_if_eq:nnTF { #1 } { inherit }
  {
    \tl_set:Nn \l__glossaries_tree_post_child_loc_tl
    {
      \l__glossaries_tree_post_loc_tl
    }
  }
  {
    \tl_set:Nn \l__glossaries_tree_post_loc_tl { #1 }
  }
} ,

```

Top-level and level 1 item counters:

```

item-counter-width .code:n =
{
  \tl_if_eq:nn { #1 } { natural }
  {
    \tl_clear:N \l__glossaries_tree_prenamesym_width_tl
  }
  {
    \tl_set:N \l__glossaries_tree_prenamesym_width_tl { #1 }
  }
},
item-counter-padding .code:n =
{
  \tl_clear_new:N \l__glossaries_tree_itemcounter_padding_tl
  \tl_set:Nn \l__glossaries_tree_itemcounter_padding_tl { #1 }
} ,
sub-item-counter-width .code:n =
{
  \tl_if_eq:nn { #1 } { natural }
  {
    \tl_clear:N \l__glossaries_tree_presubnamesym_width_tl
  }
  {
    \tl_set:N \l__glossaries_tree_presubnamesym_width_tl { #1 }
  }
},
sub-item-counter-padding .code:n =
{
  \tl_clear_new:N \l__glossaries_tree_itemcounteri_padding_tl
  \tl_set:Nn \l__glossaries_tree_itemcounteri_padding_tl { #1 }
} ,
item-counter-align .choice: ,
item-counter-align / l .code:n =
{ \tl_set:Nn \l__glossaries_tree_prenamesym_align_tl { l } } ,
item-counter-align / r .code:n =

```

```

{ \tl_set:Nn \l__glossaries_tree_prenamesym_align_tl { r } } ,
item-counter-align / c .code:n =
{ \tl_set:Nn \l__glossaries_tree_prenamesym_align_tl { c } } ,
sub-item-counter-align .choice: ,
sub-item-counter-align / l .code:n =
{ \tl_set:Nn \l__glossaries_tree_presubnamesym_align_tl { l } } ,
sub-item-counter-align / r .code:n =
{ \tl_set:Nn \l__glossaries_tree_presubnamesym_align_tl { r } } ,
sub-item-counter-align / c .code:n =
{ \tl_set:Nn \l__glossaries_tree_presubnamesym_align_tl { c } } ,
item-counter-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set_eq:NN \__glossaries_tree_prenamesym_font:n \use:n
  }
  {
    \cs_set:Nn \__glossaries_tree_prenamesym_font:n
    {
      { #1 { ##1 } }
    }
  }
} ,
sub-item-counter-font .code:n =
{
  \tl_if_empty:nTF { #1 }
  {
    \cs_set_eq:NN \__glossaries_tree_presubnamesym_font:n \use:n
  }
  {
    \cs_set:Nn \__glossaries_tree_presubnamesym_font:n
    {
      { #1 { ##1 } }
    }
  }
} ,

```

Name width (in the unlikely event that deeper levels are required, the low-level command will need to be explicitly used instead):

```

name-width .code:n =
{
  \glossaries_tree_set_name_width:nn { 0 } { #1 }
} ,
sub-name-width .code:n =
{
  \glossaries_tree_set_name_width:nn { 1 } { #1 }
} ,
sub-sub-name-width .code:n =
{
  \glossaries_tree_set_name_width:nn { 2 } { #1 }
}

```

```

    } ,
name-align .choice: ,
name-align / l .code:n =
{ \tl_set:Nn \l__glossaries_tree_namealign_tl { l } } ,
name-align / r .code:n =
{ \tl_set:Nn \l__glossaries_tree_namealign_tl { r } } ,
name-align / c .code:n =
{ \tl_set:Nn \l__glossaries_tree_namealign_tl { c } } ,
widest-name .code:n =
{
  \glossaries_tree_set_widest_name:nn { 0 } { #1 }
} ,
widest-sub-name .code:n =
{
  \glossaries_tree_set_widest_name:nn { 1 } { #1 }
} ,
widest-sub-sub-name .code:n =
{
  \glossaries_tree_set_widest_name:nn { 2 } { #1 }
} ,
update-widest-name .code:n =
{
  \glossaries_tree_update_widest_name:nn { 0 } { #1 }
} ,
update-widest-sub-name .code:n =
{
  \glossaries_tree_update_widest_name:nn { 1 } { #1 }
} ,
update-widest-sub-sub-name .code:n =
{
  \glossaries_tree_update_widest_name:nn { 2 } { #1 }
} ,

```

Symbol width (in the unlikely event that deeper levels are required, the low-level command will need to be explicitly used instead):

```

symbol-width .code:n =
{
  \glossaries_tree_set_symbol_width:nn { 0 } { #1 }
} ,
sub-symbol-width .code:n =
{
  \glossaries_tree_set_symbol_width:nn { 1 } { #1 }
} ,
sub-sub-symbol-width .code:n =
{
  \glossaries_tree_set_symbol_width:nn { 2 } { #1 }
} ,
symbol-align .choice: ,
symbol-align / l .code:n =
{ \tl_set:Nn \l__glossaries_tree_symbolalign_tl { l } } ,

```



```

symbol-align / r .code:n =
{ \tl_set:Nn \l__glossaries_tree_symbolalign_tl { r } } ,
symbol-align / c .code:n =
{ \tl_set:Nn \l__glossaries_tree_symbolalign_tl { c } } ,
widest-symbol .code:n =
{
  \glossaries_tree_set_widest_symbol:nn { 0 } { #1 }
} ,
widest-sub-symbol .code:n =
{
  \glossaries_tree_set_widest_symbol:nn { 1 } { #1 }
} ,
widest-sub-sub-symbol .code:n =
{
  \glossaries_tree_set_widest_symbol:nn { 2 } { #1 }
} ,
update-widest-symbol .code:n =
{
  \glossaries_tree_update_widest_symbol:nn { 0 } { #1 }
} ,
update-widest-sub-symbol .code:n =
{
  \glossaries_tree_update_widest_symbol:nn { 1 } { #1 }
} ,
update-widest-sub-sub-symbol .code:n =
{
  \glossaries_tree_update_widest_symbol:nn { 2 } { #1 }
} ,

```

Combined width of name and symbol and separators.

```

name-symbol-width .code:n =
{
  \glossaries_tree_set_name_symbol_width:nn { 0 } { #1 }
} ,
sub-name-symbol-width .code:n =
{
  \glossaries_tree_set_name_symbol_width:nn { 1 } { #1 }
} ,
sub-sub-name-symbol-width .code:n =
{
  \glossaries_tree_set_name_symbol_width:nn { 2 } { #1 }
} ,
name-symbol-align .choice: ,
name-symbol-align / l .code:n =
{ \tl_set:Nn \l__glossaries_tree_namesymbolalign_tl { l } } ,
name-symbol-align / r .code:n =
{ \tl_set:Nn \l__glossaries_tree_namesymbolalign_tl { r } } ,
name-symbol-align / c .code:n =
{ \tl_set:Nn \l__glossaries_tree_namesymbolalign_tl { c } } ,
}

```

Style variables:

```
\tl_new:N \l__glossaries_tree_par_setup_tl
\tl_new:N \l__glossaries_tree_group_title_tl
\int_new:N \l__glossaries_tree_level_int
```

Style definition:

```
\newglossarystyle{tree*}
{
  \renewenvironment { theglossary }
  { \__glossaries_tree_begin: }
  { \__glossaries_tree_end: }
  \cs_set_eq:NN \glossaryheader \__glossaries_tree_header:
  \cs_set_eq:NN \glsgroupheading \__glossaries_tree_group_heading:n
  \cs_set_eq:NN \glssubgroupheading \__glossaries_tree_group_heading:nnn
  \cs_set_eq:NN \glossentry \__glossaries_tree_item:nn
  \cs_set_eq:NN \subglossentry \__glossaries_tree_subitem:nnn
  \tl_set:Nn \glsgroupskip
  {
    \legacy_if:nF { glsnogroupskip }
    { \par \skip_vertical:n { \l__glossaries_tree_group_skip_tl } }
  }
}
```

Implement group bookmark if available.

```
\cs_new:Nn \__glossaries_tree_bookmark_group:nnn
{
  \tl_if_exist:cTF { toplevel@ \@@glossarysec }
  {
    \exp_args:Ne \glossaries_tree_bookmark_group:nnn
    {
      \int_eval:n
      {
        \tl_use:c { toplevel@ \@@glossarysec } + \c_one_int + #1
      }
    }
    { #2 } { #3 }
  }
  {
    \exp_args:Ne \glossaries_tree_bookmark_group:nnn
    { \int_eval:n { \c_one_int + #1 } }
    { #2 } { #3 }
  }
}
\cs_new:Nn \__glossaries_tree_pregrouphead:nnn { }
```

Group heading:

```
\cs_new:Nn \__glossaries_tree_group_heading:n
{
  \bool_if:NTF \l__glossaries_tree_group_headings_bool
  {
    \cs_if_exist:NTF \glstrgetgrouptitle
```

```

    {
      \glstrgetgrouptitle { #1 } \l__glossaries_tree_group_title_tl
    }
    {
      \@gls@getgrouptitle { #1 } \l__glossaries_tree_group_title_tl
    }
    \exp_args:NnV \__glossaries_tree_group_heading:nnn { #1 }
    \l__glossaries_tree_group_title_tl
    { \c_zero_int }
    \glswriteglossarygroup { #1 } { \l__glossaries_tree_group_title_tl }
  }
  {
    \int_zero:N \l__glossaries_tree_level_int
  }
}

```

Sub group heading:

```

\cs_new:Nn \__glossaries_tree_group_heading:nnnn
{
  \bool_if:NTF \l__glossaries_tree_group_headings_bool
  {
    \cs_if_exist:NTF \glstrgetgrouptitle
    {
      \glstrgetgrouptitle { #4 } \l__glossaries_tree_group_title_tl
    }
    {
      \@gls@getgrouptitle { #4 } \l__glossaries_tree_group_title_tl
    }
  }
  \exp_args:Nne \__glossaries_tree_group_heading:nnn { #1 }
  {
    \glossaries_tree_subgroup_title:nn { #3 }
    { \exp_not:V \l__glossaries_tree_group_title_tl }
  }
  { #2 }
  \glswriteglossarysubgroup
  { #2 } { #3 } { #4 }
  { \l__glossaries_tree_group_title_tl }
}
{
  \int_set:Nn \l__glossaries_tree_level_int { #2 }
}
}

```

Lower-level group heading with label, title and level argument.

```

\cs_new:Nn \__glossaries_tree_group_heading:nnn
{
  \par
  \group_begin:
  \everypar { }
  \dim_zero:N \parindent
  \dim_zero:N \hangindent
}

```

```

\dim_zero:N \parskip
\int_set:Nn \l__glossaries_tree_level_int { #3 }
\__glossaries_tree_pregrouphead:nnn
{ \l__glossaries_tree_level_int } { #1 } { #2 }
\int_if_zero:nTF { \l__glossaries_tree_level_int }
{
  \tl_set:Nn \l__glossaries_tree_group_title_tl
  {
    \__glossaries_tree_header_font:n { #2 }
  }
}
{
  \tl_set:Nn \l__glossaries_tree_group_title_tl
  {
    \__glossaries_tree_sub_header_font:n { #2 }
  }
}
\bool_if:NT \l__glossaries_tree_hyprnav_bool
{
  \tl_set:Nn \l__glossaries_tree_group_title_tl
  {
    \exp_not:N \glsnavhypertarget { #1 }
    { \exp_not:N \l__glossaries_tree_group_title_tl }
  }
}
\int_if_zero:nTF { \l__glossaries_tree_level_int }
{
  \__glossaries_tree_header_fmt:n
  {
    \l__glossaries_tree_group_title_tl
  }
}
{
  \glossaries_tree_pre_subgroup:nn { #1 } { #3 }
  \__glossaries_tree_sub_header_fmt:n
  {
    \l__glossaries_tree_group_title_tl
  }
}
\nopagebreak
\tl_if_empty:NF \l__glossaries_tree_postgrouphead_skip_tl
{
  \skip_vertical:n { \l__glossaries_tree_postgrouphead_skip_tl }
}
\par
\@afterheading
\expandafter
\group_end:
\if@endpe \@doendpe \fi
}

```

Start of glossary.

```

\cs_new:Nn \l__glossaries_tree_begin:
{
  \tl_if_empty:NF \l__glossaries_tree_offset_tl
  {
    \dim_set:Nn \l__glossaries_tree_offset_dim
      { \l__glossaries_tree_offset_tl }
  }
  \__glossaries_tree_init_namesymbolwidths:
  \bool_if:NT \l__glossaries_tree_hide_groups_bool
  {
    \bool_set_false:N \l__glossaries_tree_group_headings_bool
    \legacy_if_set_true:n { glsnogroupskip }
  }
  \int_set:Nn \l__glossaries_tree_level_int { -1 }
  \tl_clear:N \l__glossaries_tree_par_setup_tl
  \tl_if_empty:NT \l__glossaries_tree_parindent_tl
  {
    \tl_if_empty:NTF \l__glossaries_tree_hangindent_tl
    {
      \tl_set:Ne \l__glossaries_tree_parindent_tl
        { \dim_use:N \parindent }
    }
    {
      \tl_set:Ne \l__glossaries_tree_parindent_tl
        { \dim_eval:n { \parindent + \l__glossaries_tree_hangindent_tl } }
    }
  }
  \tl_put_right:Nn \l__glossaries_tree_par_setup_tl
  {
    \dim_set:Nn \parindent
    {
      \l__glossaries_tree_level_int
      \l__glossaries_tree_offset_dim
      + \l__glossaries_tree_parindent_tl
    }
  }
  \tl_if_empty:NF \l__glossaries_tree_parskip_tl
  {
    \tl_put_right:Nn \l__glossaries_tree_par_setup_tl
    {
      \skip_set:Nn \parskip { \l__glossaries_tree_parskip_tl }
    }
  }
  \tl_if_empty:NT \l__glossaries_tree_hangindent_tl
  {
    \tl_set:Nn \l__glossaries_tree_hangindent_tl { \c_zero_dim }
  }
  \tl_put_right:Nn \l__glossaries_tree_par_setup_tl

```

```

{
  \everypar
  {
    \int_compare:nNnF { \l__glossaries_tree_level_int } < { \c_zero_int }
    {
      \dim_set:Nn \hangindent
      {
        \l__glossaries_tree_level_int
        \l__glossaries_tree_offset_dim
        + \l__glossaries_tree_hangindent_tl
      }
    }
  }
}

}

End of glossary. (Postamble comes after this.)
\cs_new:Nn \__glossaries_tree_end: { \strut \par }

Top-level item:
\cs_new:Nn \__glossaries_tree_item:nn
{
  \int_zero:N \l__glossaries_tree_level_int
  \dim_set:Nn \parindent { \c_zero_dim }
  \par
  \dim_set:Nn \hangindent
  {
    \l__glossaries_tree_hangindent_tl
  }
  \glossaries_tree_pre_item:nnn { 0 } { #1 } { #2 }
  \__glossaries_tree_namesymbox:n
  {
    \__glossaries_tree_prenamesym:n
    {
      \__glossaries_tree_prenamesym_font:n
      {
        \glentryitem { #1 }
      }
    }
  }
  \ifglshassymbol { #1 }
  {
    \__glossaries_tree_name_sym_item:nnn
    { #1 }
    {
      \__glossaries_tree_namebox:n
      {
        \__glossaries_tree_name_font:n
        {
          \__glossaries_tree_name_encap:n
          { \__glossaries_tree_gloss_name:n { #1 } }
        }
      }
    }
  }
}

```

```

    }
  }
  {
    \__glossaries_tree_symbolbox:n
    {
      \__glossaries_tree_symbol_font:n
      {
        \__glossaries_tree_symbol_encap:n
        { \__glossaries_tree_gloss_symbol:n { #1 } }
      }
    }
  }
}
{
  \__glossaries_tree_name_sym_item:nnn
  { #1 }
  {
    \__glossaries_tree_namebox:n
    {
      \__glossaries_tree_name_font:n
      {
        \__glossaries_tree_name_encap:n
        { \__glossaries_tree_gloss_name:n { #1 } }
      }
    }
  }
}
{ }
}
\__glossaries_tree_show_desc:nn { #1 }
{
  \l__glossaries_tree_post_namesymbol_tl
}
}
\l__glossaries_tree_par_setup_tl
\__glossaries_tree_show_desc:nn { #1 }
{
  \__glossaries_tree_desc_font:n
  {
    \l__glossaries_tree_pre_desc_tl
    \__glossaries_tree_gloss_desc:n { #1 }
  }
}
\__glossaries_tree_show_loc:nnn { #1 } { #2 }
{
  \l__glossaries_tree_pre_loc_tl
  \__glossaries_tree_loc_font:n { #2 }
  \l__glossaries_tree_post_loc_tl
}
\glossaries_tree_post_item:nnn { 0 } { #1 } { #2 }
}

```

```

\cs_new:Nn \__glossaries_tree_subitem:nnn
{
  \int_set:Nn \l__glossaries_tree_level_int { #1 }
  \dim_set:Nn \parindent
  {
    \l__glossaries_tree_level_int
    \l__glossaries_tree_offset_dim
  }
  \par
  \dim_set:Nn \hangindent
  {
    \l__glossaries_tree_level_int
    \l__glossaries_tree_offset_dim
    + \l__glossaries_tree_hangindent_tl
  }
  \glossaries_tree_pre_item:nnn { #1 } { #2 } { #3 }
  \exp_args:Ne \__glossaries_tree_namesymbox:nn
  { \romannumeral #1 }
  {
    \int_compare:nNnT { #1 } = { \c_one_int }
    {
      \__glossaries_tree_presubnamesym:n
      {
        \__glossaries_tree_presubnamesym_font:n
        {
          \glssubentryitem { #2 }
        }
      }
    }
  }
  \ifglshassymbol { #2 }
  {
    \__glossaries_tree_name_sym_subitem:nnn
    { #2 }
    {
      \exp_args:Ne \__glossaries_tree_namebox:nn
      { \romannumeral #1 }
      {
        \__glossaries_tree_child_name_font:n
        {
          \__glossaries_tree_child_name_encap:n
          { \__glossaries_tree_child_gloss_name:n { #2 } }
        }
      }
    }
  }
  {
    \exp_args:Ne \__glossaries_tree_symbolbox:nn
    { \romannumeral #1 }
    {
      \__glossaries_tree_child_symbol_font:n
      {

```



```

        \_glossaries_tree_child_symbol_encap:n
        { \_glossaries_tree_child_gloss_symbol:n { #2 } }
    }
}
}
{
\_glossaries_tree_name_sym_subitem:nnn
{ #2 }
{
    \exp_args:Ne \_glossaries_tree_namebox:nn
    { \romannumeral #1 }
    {
        \_glossaries_tree_child_name_font:n
        {
            \_glossaries_tree_child_name_encap:n
            { \_glossaries_tree_child_gloss_name:n { #2 } }
        }
    }
}
{ }
}
\_glossaries_tree_show_child_desc:nn { #2 }
{
    \l_glossaries_tree_post_child_namesymbol_tl
}
}
\l_glossaries_tree_par_setup_tl
\_glossaries_tree_show_child_desc:nn { #2 }
{
    \_glossaries_tree_child_desc_font:n
    {
        \l_glossaries_tree_pre_child_desc_tl
        \_glossaries_tree_child_gloss_desc:n { #2 }
    }
}
\_glossaries_tree_show_child_loc:nnn { #1 } { #3 }
{
    \l_glossaries_tree_pre_child_loc_tl
    \_glossaries_tree_child_loc_font:n { #3 }
    \l_glossaries_tree_post_loc_tl
}
}
\_glossaries_tree_post_item:nnn { #1 } { #2 } { #3 }
}

```

Navigation strip, if applicable.

```

\cs_new:Nn \_glossaries_tree_header:
{
    \bool_if:NT \l_glossaries_tree_hypernav_bool
    {

```

```

\group_begin:
\everypar { }
\dim_zero:N \parindent
\__glossaries_tree_hypernav_fmt:n
{
  \__glossaries_tree_hypernav_font:n { \glsnavigation }
  \par
}
\skip_vertical:n { \l__glossaries_tree_hypernav_skip_tl }
\@afterheading
\expandafter
\group_end:
\if@endpe \doendpe \fi
}
}
\ExplSyntaxOff

```

`\indexspace` There are a few classes that don't define `\indexspace`, so provide a definition if it hasn't been defined.

```

\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}

```

`\glstreenamefmt` Format used to display the name in the tree styles. (This may be counteracted by `\glsnamefont`.) This command was previously also used to format the group headings.

```

\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}

```

`\glstreegroupheaderfmt` Format used to display the group header in the tree styles. Before v4.22, `\glstreenamefmt` was used for the group header, so the default definition uses that to help maintain backward-compatibility, since in previous versions redefining `\glstreenamefmt` would've also affected the group headings.

```

\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}

```

`\glstreenavigationfmt` Format used to display the navigation header in the tree styles.

```

\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}

```

Allow the user to adjust the index style without disturbing the index.

`\glstreeitem` Top level item used in index style.

```

\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}

```

`\glstreesubitem` Level 1 item used in index style.

```

\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}

```

`\glstreesubsubitem` Level 1 item used in index style.

```
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
```

`\glstreepredesc` Allow the user to adjust the space before the description (except for the `alttree` style).

```
\newcommand{\glstreepredesc}{\space}
```

`\glstreechildpredesc` Allow the user to adjust the space before the description for sub-entries (except for the `treenoname` and `alttree` style).

```
\newcommand{\glstreechildpredesc}{\space}
```

`index` The index glossary style is similar in style to the way indices are usually typeset using `\item`, `\subitem` and `\subsubitem`. The entry name is set in bold. If an entry has a symbol, it is placed in brackets after the name. Then the description is displayed, followed by the number list. This style allows up to three levels.

```
\newglossarystyle{index}{%
```

Set the paragraph indentation and skip and define `\item` to be the same as that used by `theindex`:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
```

Do nothing at the start of the environment:

```
\renewcommand*{\glossaryheader}{}%
```

No group headers:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entry starts a new item with the name in bold followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand*{\glossentry}[2]{%
\item\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
  \glstentryitem{##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
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}

```

\glstreeindent Length governing left indent for each level of the `tree` style.

```
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}

```

treenoname The `treenoname` glossary style is like the `tree` style, but doesn't print the name or symbol for sub-levels.

```
\newglossarystyle{treenoname}{%
```

Set the paragraph indentation and skip:

```
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%

```

No header:

```
\renewcommand*{\glossaryheader}{}%
```

No group headings:

```
\renewcommand*{\glsgroupheading}[1]{}%
```

Main (level 0) entries: the name is in bold, followed by the symbol in brackets (if it exists), the description and the page list.

```
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\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
  {\glsggetgrouptitle{##1}}\par\indexspace}%
}
```

treenonamehypergroup The `treenonamehypergroup` style is like the `treenonamegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{treenonamehypergroup}{%
```

Base it on the `glostyletreenoname` style:

```
\setglossarystyle{treenoname}%
```

Put navigation links to the groups at the start of the `theglossary` environment:

```
\renewcommand*{\glossaryheader}{%
  \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}
```

Each group has a heading (in bold with a target) followed by a vertical gap):

```
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt
  {\glsnavhypertarget{##1}{\glsggetgrouptitle{##1}}}\par
  \indexspace}%
}
```

\glsfindwidesttoplevelname Find the widest name over all parentless entries in the given glossary or glossaries.

```
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
```

```

\gls@tmplen=0pt\relax
\forall glossaries[#1]{\@gls@type}%
{%
  \forall glsentries[\@gls@type]{\@glo@label}%
  {%
    \ifgls hasparent{\@glo@label}%
    {}%
    {%
      \glsmeasurewidth{\dimen@}%
      {\glstreenamefmt{\glsentryname{\@glo@label}}}%
      \ifdim\dimen@>\gls@tmplen
        \gls@tmplen=\dimen@
        \letcs{\@glswidestname}{glo@\glsdetoklabel{\@glo@label}@name}%
      \fi
    }%
  }%
}%
}

```

`\glssetwidest` `\glssetwidest[level]{text}` sets the widest text for the given level. It is used by the `alttree` glossary styles to determine the indentation of each level and also by the `tree*` style with applicable options. This is now just a user-level interface to the new low-level function.

```

\ExplSyntaxOn
\NewDocumentCommand \glssetwidest { 0{0} m }
{
  \glossaries_tree_set_widest_name:nn { #1 } { #2 }
}
\ExplSyntaxOff

```

`\@glswidestname` Initialise `\@glswidestname`.
`\newcommand*{\@glswidestname}{}`

`\glstreenamebox` Used by the `alttree` style to create the box for the name and associated information.
`\newcommand*{\glstreenamebox}[2]{%`
`\makebox[#1][1]{#2}%`
`}`

alttree The `alttree` glossary style is similar in style to the `tree` style, but the indentation is obtained from the width of `\@glswidestname` which is set using `\glssetwidest`.

```

\newglossarystyle{alttree}{%
  \Redefine theglossary environment.
  \renewenvironment{theglossary}%
  {\def\@gls@prevlevel{-1}%
   \mbox{}\par}%
  {\par}%
}

```


Set the header and group headers to nothing.

```
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
```

Redefine the way that the level 0 entries are displayed.

```
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
```

Find out how big the indentation should be by measuring the widest entry.

```
    \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
```

Set the hangindent and paragraph indent.

```
\hangindent\glstreeindent
\parindent\glstreeindent
```

Put the name to the left of the paragraph block.

```
\makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##1}\glspostdescription \space ##2\par
```

Set the previous level to 0.

```
\def\@gls@prevlevel{0}%
}%
```

Redefine the way sub-entries are displayed.

```
\renewcommand{\subglossentry}[3]{%
```

Increment and display the sub-entry counter if this is a level 1 entry and the sub-entry counter is in use.

```
\ifnum##1=1\relax
  \glssubentryitem{##2}%
\fi
```

If the level hasn't changed, keep the same settings, otherwise adjust `\glstreeindent` accordingly.

```
\ifnum\@gls@prevlevel=##1\relax
\else
```

Compute the widest entry for this level, or for level 0 if not defined for this level.

Store in `\gls@tmplen`

```
\@ifundefined{@glswidestname\romannumeral##1}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
  \glsmeasurewidth{\gls@tmplen}{\glstreenamefmt{%
    \csname @glswidestname\romannumeral##1\endcsname\space}}}%
```

Determine if going up or down a level

```
\ifnum\@gls@prevlevel<##1\relax
```

Depth has increased, so add the width of the widest entry to `\glstreeindent`.

```
\setlength\glstreeindent\gls@tmplen
\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
```

Depth has decreased, so subtract width of the widest entry from the previous level to `\glstreeindent`. First determine the width of the widest entry for the previous level and store in `\glstreeindent`.

```
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\glsmeasurewidth{\glstreeindent}{\glstreenamfmt{%
\@glswidestname\space}}}{%
\glsmeasurewidth{\glstreeindent}{\glstreenamfmt{%
\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]{\glstreenambox{\gls@tmplen}{%
\glstreenamfmt{\glstarget{##2}{\glossentryname{##2}}}}}%
```

If the symbol is missing, ignore it, otherwise put it in brackets.

```
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
```

Do the description followed by the description terminator and location list.

```
\glossentrydesc{##2}\glspostdescription\space ##3\par
```

Set the previous level macro to the current level.

```
\def\@gls@prevlevel{##1}%
}%
```

Vertical gap between groups is the same as that used by indices:

```
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
```

alttreegroup Like the `alttree` style but the glossary groups have headings.

```
\newglossarystyle{alttreegroup}{%
```

Base it on the `glostylealttree` style:

```
\setglossarystyle{alttree}%
```

Give each group a heading.

```
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
```

```

\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}

```

alttreehypergroup The `alttreehypergroup` style is like the `alttreegroup` style, but has a set of links to the groups at the start of the glossary.

```
\newglossarystyle{alttreehypergroup}{%
```

Base it on the `glostylealttree` style:

```
\setglossarystyle{alttree}%
```

Put the navigation links in the header

```

\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%

```

Put a hypertarget at the start of each group

```

\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

4 Multi-Lingual Support

4.1 Babel Captions (`glossaries-babel.sty`)

Many thanks to everyone who contributed to the translations both via email and on `comp.text.tex`. Language support has now been split off into independent language modules.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-babel-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-babel-2021-11-01.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}

```

Declare package

```
\ProvidesPackage{glossaries-babel}[2026/04/21 v5.1 (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}%
    }%
  }%
}%
{}%
```

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{v5.1}{2026-04-21}
```

Declare package

```
\ProvidesPackage{glossaries-polyglossia}[2026/04/21 v5.1 (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{v5.1}{2026-04-21}

```

Declare package:

```
\ProvidesPackage{glossaries-prefix}[2026/04/21 v5.1 (NLCT)]
```

Pass all options to glossaries:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

Load glossaries:

```
\RequirePackage{glossaries}
```

Add the new keys:

```

\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%

```

Add them to \@gls@keymap:

```

\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}

```

Set the default values:

```

\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
}

```

```

\let\@glo@entryprefixfirstplural\@gls@default@value
}

```

Set the assignment code:

```

\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label}{prefix}{\@glo@entryprefix}%
\gls@assign@field{}\@glo@label}{prefixplural}{\@glo@entryprefixplural}%

```

If prefixfirst has not been supplied, make it the same as prefix.

```

\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}%
{\@glo@entryprefixfirst}%

```

If prefixfirstplural has not been supplied, make it the same as prefixplural.

```

\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefixplural\endcsname}{\@glo@label}%
{prefixfirstplural}{\@glo@entryprefixfirstplural}%
}

```

Define commands to access these fields:

```

\glsentryprefixfirst
\newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}

```

```

\glsentryprefixfirstplural
\newcommand*{\glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}

```

```

\glsentryprefix
\newcommand*{\glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}

```

```

\glsentryprefixplural
\newcommand*{\glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}

```

Now for the initial upper case variants:

```

\Glsentryprefixfirst
\newcommand*{\Glsentryprefixfirst}[1]{%
\@Glsentryfield{#1}{prefixfirst}%
}

```

Add sentence case mapping.

```

\glsmfuaddmap{\glsentryprefixfirst}{\Glsentryprefixfirst}

```

```

\Glsentryprefixfirstplural
\newcommand*{\Glsentryprefixfirstplural}[1]{%
\@Glsentryfield{#1}{prefixfirstplural}%
}

```

Add sentence case mapping.

```

\glsmfuaddmap{\glsentryprefixfirstplural}{\Glsentryprefixfirstplural}

```

`\Glsentryprefix`

```
\newcommand*{\Glsentryprefix}[1]{%
  \@Glsentryfield{#1}{prefix}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefix}{\Glsentryprefix}
```

`\Glsentryprefixplural`

```
\newcommand*{\Glsentryprefixplural}[1]{%
  \@Glsentryfield{#1}{prefixplural}%
}
```

Add sentence case mapping.

```
\glsmfuaddmap{\glsentryprefixplural}{\Glsentryprefixplural}
```

Define commands to determine if the prefix keys have been set:

`\ifglshasprefix`

```
\newcommand*{\ifglshasprefix}[3]{%
  \ifcempty{glo@\glsdetoklabel{#1}@prefix}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixplural`

```
\newcommand*{\ifglshasprefixplural}[3]{%
  \ifcempty{glo@\glsdetoklabel{#1}@prefixplural}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixfirst`

```
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@\glsdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
```

`\ifglshasprefixfirstplural`

```
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@\glsdetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
```

`\@glsprefix@record@hook` Need to take into account the possibility that glossaries-extra might be loaded with the record option.

```
\providecommand{\@glsprefix@record@hook}[2]{%
```

```

\ifdef\@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}%
    {%
      \ifgls@hasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirst{#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}%
  \glsoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefixplural{#2}{\glstryentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirstplural{#2}%
      {\glstryentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glsp1@{#1}{#2}[#3]%
  }%
}
```

Now for the first letter upper case versions:

```
\PglS
  \newrobustcmd{\PglS}{\@glshyp@opt\@PglS}
```

Add sentence case mapping.

```
\glsmfuaddmap{\pgls}{\PglS}
```

\@PglS Unstarred version.

```
\newcommand*{\@PglS}[2][{}]{%
  \new@ifnextchar[%
  {\@PglS@{#1}{#2}}%
  {\@PglS@{#1}{#2}[]}%
}
```

\@PglS@ Read in the final optional argument:

```
\def\@PglS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {%
        \Glstryentryprefix{#2}%
        \glsprefixsep
        \@glS@{#1}{#2}[#3]%
      }%
      {\@GlS@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirst{#2}%

```

```

    {%
      \Glsentryprefixfirst{#2}%
      \glsprefixsep
      \@gls@{#1}{#2}[#3]%
    }%
    {\@Gls@{#1}{#2}[#3]}%
  }%
}

```

Similarly for the plural version:

```

\Pglspl
\newrobustcmd{\Pglspl}{\@gls@hyp@opt\@Pglspl}

```

Add sentence case mapping.

```

\glsmfuaddmap{\pglspl}{\Pglspl}

```

\@Pglspl Unstarred version.

```

\newcommand*{\@Pglspl}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglspl@{#1}{#2}}%
    {\@Pglspl@{#1}{#2}[]}%
  }

```

\@Pglspl@ Read in the final optional argument:

```

\def\@Pglspl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \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]{%
\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
\ifglsused{#2}%
{%
\ifgls@hasprefix{#2}%
{\gls@supercase{\gls@entryprefix{#2}\gls@prefixsep}}{}}%
}%
{%
\ifgls@hasprefixfirst{#2}%
{\gls@supercase{\gls@entryprefixfirst{#2}\gls@prefixsep}}{}}%
}%
\@GSL@{#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]{%
\@glsprefix@record@hook{#1}{#2}%
}
```

```

\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \ifglshasprefixplural{#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{v5.1}{2026-04-21}

```

Package version number now in line with main `glossaries` package number.

```

\ProvidesPackage{glossaries-accsupp}[2026/04/21 v5.1 (NLCT)
Experimental glossaries accessibility]

```

Pass all options to `glossaries`:

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
```

Process options:

```
\ProcessOptions
```

This package should be loaded before `glossaries-extra`, so complain if that has already been loaded.

```

\@ifpackageloaded{glossaries-extra}
{%

```

If the `accsupp` option was used, `\@glsxtr@doaccsupp` will have been set, otherwise it will be empty.

```

\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The ‘glossaries-accsupp’
package has been loaded\MessageBreak
after the ‘glossaries-extra’ package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the ‘accsupp’ option when you load\MessageBreak

```

```

        'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
        before loading 'glossaries-extra'%
    \fi
}
{}

```

`\compatibleglossentry` Override style compatibility macros:

`\compatiblesubglossentry`

Required packages:

```
\RequirePackage{glossaries}
```

`\gls@accsupp@engine` There's currently only support for accsupp, but if you define `\gls@accsupp@engine` before loading `glossaries-accsupp`, you can prevent `accsupp` from being loaded. Redefining this command after `glossaries-accsupp` has loaded obviously won't do anything (and so is an internal command to deter casual use). If it is defined to something other than `accsupp` then `\gls@accessibility` will need to be defined to something appropriate.

```
\providecommand{\gls@accsupp@engine}{accsupp}
```

```

\gls@accessibility{<options>}{<PDF element>}{<value>}{<content>}}

```

`\gls@accessibility`

```

\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
  }
}
{}

```

```

\glsaccessibility[<options>]{<PDF element>}{<value>}{<content>}}

```

`\glsaccessibility`

User-level command that includes debug info if required.

```

\newcommand{\glsaccessibility}[4][[]]{%
  \@glsshowaccsupp{#1}{#2}{#3}%
  \gls@accessibility{#1}{#2}{#3}{#4}%
}

```

6.1 Defining Replacement Text

The version 0.1 stored the replacement text in the `symbol` key. This has been changed to use the new keys defined here. Example of use:

```
%\newglossaryentry{dr}{name=Dr,description={},access={Doctor}}
%
```

`access` The replacement text corresponding to the `name` key:

```
\define@key{glossentry}{access}{%
\def\@glo@access{#1}%
}
```

`textaccess` The replacement text corresponding to the `text` key:

```
\define@key{glossentry}{textaccess}{%
\def\@glo@textaccess{#1}%
}
```

`firstaccess` The replacement text corresponding to the `first` key:

```
\define@key{glossentry}{firstaccess}{%
\def\@glo@firstaccess{#1}%
}
```

`pluralaccess` The replacement text corresponding to the `plural` key:

```
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
```

`firstpluralaccess` The replacement text corresponding to the `firstplural` key:

```
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
```

`symbolaccess` The replacement text corresponding to the `symbol` key:

```
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
```

`symbolpluralaccess` The replacement text corresponding to the `symbolplural` key:

```
\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}
```

`descriptionaccess` The replacement text corresponding to the `description` key:

```
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
```

descriptionpluralaccess The replacement text corresponding to the descriptionplural key:

```
\define@key{glossentry}{descriptionpluralaccess}{%  
  \def\@glo@descpluralaccess{#1}%  
}
```

shortaccess The replacement text corresponding to the short key:

```
\define@key{glossentry}{shortaccess}{%  
  \def\@glo@shortaccess{#1}%  
}
```

shortpluralaccess The replacement text corresponding to the shortplural key:

```
\define@key{glossentry}{shortpluralaccess}{%  
  \def\@glo@shortpluralaccess{#1}%  
}
```

longaccess The replacement text corresponding to the long key:

```
\define@key{glossentry}{longaccess}{%  
  \def\@glo@longaccess{#1}%  
}
```

longpluralaccess The replacement text corresponding to the longplural key:

```
\define@key{glossentry}{longpluralaccess}{%  
  \def\@glo@longpluralaccess{#1}%  
}
```

There are now also keys that correspond to the user keys:

user1access The replacement text corresponding to the user1 key:

```
\define@key{glossentry}{user1access}{%  
  \def\@glo@useriaccess{#1}%  
}
```

user2access The replacement text corresponding to the user2 key:

```
\define@key{glossentry}{user2access}{%  
  \def\@glo@useriiaccess{#1}%  
}
```

user3access The replacement text corresponding to the user3 key:

```
\define@key{glossentry}{user3access}{%  
  \def\@glo@useriiiaccess{#1}%  
}
```

user4access The replacement text corresponding to the user4 key:

```
\define@key{glossentry}{user4access}{%  
  \def\@glo@userivaccess{#1}%  
}
```

`user5access` The replacement text corresponding to the `user5` key:

```
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
```

`user6access` The replacement text corresponding to the `user6` key:

```
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
```

For any custom keys, the replacement text would have to be explicitly put in the value, e.g., `user1={\glsshortaccsupp{inches}{in}}`.

Append these new keys to `\@gls@keymap`:

```
\appto\@gls@keymap{,%
{access}{access},%
{textaccess}{textaccess},%
{firstaccess}{firstaccess},%
{pluralaccess}{pluralaccess},%
{firstpluralaccess}{firstpluralaccess},%
{symbolaccess}{symbolaccess},%
{symbolpluralaccess}{symbolpluralaccess},%

{descriptionaccess}{descaccess},%
{descriptionpluralaccess}{descpluralaccess},%
{shortaccess}{shortaccess},%
{shortpluralaccess}{shortpluralaccess},%
{longaccess}{longaccess},%
{longpluralaccess}{longpluralaccess},%
{user1access}{useriaccess},%
{user2access}{useriiaccess},%
{user3access}{useriiiaccess},%
{user4access}{userivaccess},%
{user5access}{uservaccess},%
{user6access}{userviaccess}%
}
```

`\@gls@noaccess` Indicates that no replacement text has been provided.

```
\def\@gls@noaccess{\relax}
```

Previously, the `access` key was initialised to the value of the `symbol` key at the start for backwards compatibility. This causes a problem for situations where the replacement text is provided for `symbol` but not for `name` so this behaviour has been removed.

```
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
\@gls@oldnewglossaryentryprehook
\def\@glo@access{\relax}%
}
```


Initialise the other keys:

```

\def\@glo@textaccess{\@glo@access}%
\def\@glo@firstaccess{\@glo@access}%
\def\@glo@pluralaccess{\@glo@textaccess}%
\def\@glo@firstpluralaccess{\@glo@pluralaccess}%
\def\@glo@symbolaccess{\relax}%
\def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
\def\@glo@descaccess{\relax}%
\def\@glo@descpluralaccess{\@glo@descaccess}%
\def\@glo@shortaccess{\relax}%
\def\@glo@shortpluralaccess{\@glo@shortaccess}%
\def\@glo@longaccess{\relax}%
\def\@glo@longpluralaccess{\@glo@longaccess}%
\def\@glo@useriaccess{\relax}%
\def\@glo@useriiaaccess{\relax}%
\def\@glo@useriiiaaccess{\relax}%
\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}

```

Add to the end hook:

```

\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
\@gls@oldnewglossaryentryposthook

```

Store the access information:

```

\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
\@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
\@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
\@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
\@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
\@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
\@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
\@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%

```

```

        \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @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*{\glentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}

```

`\glentrypluralaccess` Get the value of the pluralaccess key for the entry with the given label:

```

\newcommand*{\glentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}

```

`\glentryfirstpluralaccess` Get the value of the firstpluralaccess key for the entry with the given label:

```

\newcommand*{\glentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}

```

`\glentrysymbolaccess` Get the value of the symbolaccess key for the entry with the given label:

```

\newcommand*{\glentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}

```

`\glentrysymbolpluralaccess` Get the value of the symbolpluralaccess key for the entry with the given label:

```

\newcommand*{\glentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}

```

`\glentrydescaccess` Get the value of the descriptionaccess key for the entry with the given label:

```

\newcommand*{\glentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}

```

`\glentrydescpluralaccess` Get the value of the descriptionpluralaccess key for the entry with the given label:

```

\newcommand*{\glentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}

```

`\glentryshortaccess` Get the value of the shortaccess key for the entry with the given label:

```

\newcommand*{\glentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}

```

`\glentryshortpluralaccess` Get the value of the shortpluralaccess key for the entry with the given label:

```

\newcommand*{\glentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}

```

`\glentrylongaccess` Get the value of the longaccess key for the entry with the given label:

```

\newcommand*{\glentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

`\glentrylongpluralaccess` Get the value of the `longpluralaccess` key for the entry with the given label:

```
\newcommand*{\glentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
```

`\glentryuseriaccess` Get the value of the `user1access` key for the entry with the given label:

```
\newcommand*{\glentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
```

`\glentryuseriiaccess` Get the value of the `user2access` key for the entry with the given label:

```
\newcommand*{\glentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
```

`\glentryuseriiiaccess` Get the value of the `user3access` key for the entry with the given label:

```
\newcommand*{\glentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
```

`\glentryuserivaccess` Get the value of the `user4access` key for the entry with the given label:

```
\newcommand*{\glentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
```

`\glentryuservaccess` Get the value of the `user5access` key for the entry with the given label:

```
\newcommand*{\glentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
```

`\glentryuserviaccess` Get the value of the `user6access` key for the entry with the given label:

```
\newcommand*{\glentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
```

There are three types of replacement text:

Alt Description of some content that’s non-textual (for example, an image). A word break is assumed after the content.

ActualText A character or sequence of characters that replaces textual content (for example, a dropped capital, a ligature or a symbol). No word break is assumed after the content.

E Expansion of an abbreviation to avoid ambiguity (for example, “St” could be short for “saint” or “street”).

Therefore, rather than having one command for all fields, it’s better to have a command dependent on the field type. For example, the `short` and `shortpl` keys would require `E`, the `symbol` key would require `ActualText`, and a field that contains an image would require `Alt`.

`\glsfieldaccsupp`

`\glsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}`

Test if there's a command called `\gls<field>accsupp`. If there is then use that otherwise use `\glsaccsupp`. The first argument should be the internal field label (not the key). The final argument is the entry label. If `glossaries-extra` has been loaded, this first checks for `\glsxtr<category><field>accsupp` and `\glsxtr<category>accsupp`.

```
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
```

`\xglsfieldaccsupp`

`\xglsfieldaccsupp{<replacement>}{<content>}{<field>}{<label>}`

As `\glsfieldaccsupp` but fully expand replacement text.

```
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
```

`\glsshortaccsupp`

`\glsshortaccsupp{<replacement text>}{<text>}`

```
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
```

`\glsshortplaccsupp`

`\glsshortplaccsupp{<replacement text>}{<text>}`

```
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
```

```
\glsaccsupp{<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}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  \expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}%
}
```

\glsnameaccessdisplay Displays the first argument with the accessibility text for the entry with the label given by the second argument (if set).

```
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@access}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
```

\glstextaccessdisplay As above but for the textaccess replacement text.

```
\newrobustcmd*{\glstextaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@textaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
```

`\glspluralaccessdisplay` As above but for the pluralaccess replacement text.

```

\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#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@glstetoklabel{#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@glstetoklabel{#2}@firstpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}

```

`\glssymbolaccessdisplay` As above but for the symbolaccess replacement text.

```

\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@symbolaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}

```

`\glssymbolpluralaccessdisplay` As above but for the symbolpluralaccess replacement text.

```

\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#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@glstetoklabel{#2}@descaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}

```

`\glsdescriptionpluralaccessdisplay` As above but for the descriptionpluralaccess replacement text.

```

\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}

```

`\glsshortaccessdisplay` As above but for the shortaccess replacement text.

```

\newrobustcmd*{\glsshortaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@shortaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{short}{#1}{\glentryshortaccess{#2}}}%
}

```

`\glsshortpluralaccessdisplay` As above but for the shortpluralaccess replacement text.

```

\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@shortpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glentryshortpluralaccess{#2}}}%
}

```

`\glslongaccessdisplay` As above but for the longaccess replacement text.

```

\newrobustcmd*{\glslongaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@longaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{long}{#1}{\glentrylongaccess{#2}}}%
}

```

`\glslongpluralaccessdisplay` As above but for the longpluralaccess replacement text.

```

\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@longpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{longpl}{#1}{\glentrylongpluralaccess{#2}}}%
}

```

`\glsuseriaccessdisplay` As above but for the user1access replacement text.

```

\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@useriaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{useri}{#1}{\glentryuseriaccess{#2}}}%
}

```

`\glsuseriiaccessdisplay` As above but for the user2access replacement text.

```

\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@useriiaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{userii}{#1}{\glentryuseriiaccess{#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}{\glentryuseriiiaccess{#2}}}%
}

```


`\glsuserivaccessdisplay` As above but for the `user4access` replacement text.

```

\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@userivaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
}

```

`\glsuservaccessdisplay` As above but for the `user5access` replacement text.

```

\newrobustcmd*{\glsuservaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@uservaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}

```

`\glsuserviaccessdisplay` As above but for the `user6access` replacement text.

```

\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
  \ifcsundef{glo@glsdetoklabel{#2}@userviaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}

```

`\glsaccessdisplay` Gets the replacement text corresponding to the named key given by the first argument and calls the appropriate command defined above.

```

\newrobustcmd*{\glsaccessdisplay}[3]{%
  \ifcsundef{gls#1accessdisplay}%
    {%
      \PackageError{glossaries-accsupp}{No accessibility support
        for key ‘#1’}{}%
    }%
    {%
      \csname gls#1accessdisplay\endcsname{#2}{#3}%
    }%
}

```

`\@@gls@default@entryfmt` Deprecated `\@@gls@default@entryfmt` was redefined to use accessibility information but has been removed in v4.50.

`\glsgenentryfmt` Redefine to use accessibility information.

```

\renewcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
    {%
      \glssifplural
    }%

```

Plural form

```

    \glscapscase
  }%

```

Don't adjust case

```
\ifglused\glslabel  
{%
```

Subsequent use

```
\glspluralaccessdisplay  
  {\glsentryplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
{%
```

First use

```
\glsfirstpluralaccessdisplay  
  {\glsentryfirstplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
}%  
{%
```

Make first letter upper case

```
\ifglused\glslabel  
{%
```

Subsequent use.

```
\glspluralaccessdisplay  
  {\Glsentryplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
{%
```

First use

```
\glsfirstpluralaccessdisplay  
  {\Glsentryfirstplural{\glslabel}}{\glslabel}%  
\glsinsert  
}%  
}%  
{%
```

Make all upper case

```
\ifglused\glslabel  
{%
```

Subsequent use

```
\glspluralaccessdisplay  
  {\glsuppercase{\glsentryplural{\glslabel}}}%  
  {\glslabel}%  
\glsuppercase{\glsinsert}%  
}%  
{%
```

First use

```
\glsfirstpluralaccessdisplay  
  {\glsuppercase{\glsentryfirstplural{\glslabel}}}%
```

```

        {\glslabel}%
      \glsuppercase{\glsinsert}%
    }%
  }%
}%
{%

```

Singular form

```

\glscapscase
{%

```

Don't adjust case

```

\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
}%
}
```

`\glsгенacfmt` Redefine to include accessibility information.

```
\renewcommand*{\glsгенacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
```

Subsequent use:

```
\glsifplural
{%
```

Subsequent plural form:

```
\glscapscase
{%
```

Subsequent plural form, don't adjust case:

```
\acronymfont
{\glsshortpluralaccessdisplay
  {\glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent plural form, make first letter upper case:

```
\acronymfont
{\glsshortpluralaccessdisplay
  {\Glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent plural form, all caps:

```

\glsuppercase
{\acronymfont
{\glsshortpluralaccessdisplay
{\glsentryshortpl{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
{%
```

Subsequent singular form

```

\glscapscase
{%
```

Subsequent singular form, don't adjust case:

```

\acronymfont
{\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent singular form, make first letter upper case:

```

\acronymfont
{\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert
}%
{%
```

Subsequent singular form, all caps:

```

\glsuppercase
{\acronymfont{%
\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
\glsinsert}%
}%
}%
{%
```

First use:

```

\glsifplural
{%
```

First use plural form:

```

\glscapscase
{%
```

First use plural form, don't adjust case:

```

\genplacrformat{\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

```
\glsupercase
{%
```

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:

```
\glsupercase
{\genacrfullformat{\glslabel}{\glsinsert}}%
}%
}%
}%
}%
}%
{%
```

User supplied text. (The insert should be empty at this point.) The accessibility information, if required, will have to be explicitly included in the custom text.

```
\glscustomtext
}%
}
```

`\genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
(\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
```

`\Genacrfullformat` Redefine to include accessibility information.

```
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
(\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
```

```

\genplacrfullformat Redefine to include accessibility information.
    \renewcommand*{\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
    }

\Genplacrfullformat Redefine to include accessibility information.
    \renewcommand*{\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
    }

\@acrshort
    \def\@acrshort#1#2[#3]{%
        \glsdoifexists{#2}%
        {%
            \let\do@gls@link@checkfirsthyper\relax

            \let\glsifplural\@secondoftwo
            \let\glscapscase\@firstofthree
            \let\glsinsert\@empty
            \def\glscustomtext{%
                \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
            }%
        }

    Call \@gls@link
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
}

\@Acrshort
    \def\@Acrshort#1#2[#3]{%
        \glsdoifexists{#2}%
        {%
            \let\do@gls@link@checkfirsthyper\relax

            \let\glsifplural\@secondoftwo
            \let\glscapscase\@secondofthree
            \let\glsinsert\@empty
            \def\glscustomtext{%
                \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
            }%
        }

    Call \@gls@link
        \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

```

```

        \glspostlinkhook
    }

\@ACRshort
\def\@ACRshort#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gl@link@checkfirsthyper\relax

        \let\glsifplural\@secondoftwo
        \let\glscapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \acronymfont{\glsshortaccessdisplay
                {\glsupercase{\glsentryshort{#2}}}{#2}}#3%
        }%
    }%
    Call \@gl@link
        \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
}

\@acrlong
\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%
        }%
    }%
    Call \@gl@link
        \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
}

\@Acrlong
\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%
}%

Call \@gls@link
  \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%

\glspostlinkhook
}

\@ACRlong
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax

    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \glsuppercase{\glsentrylong{#2}}}{#2}}#3}%
    }%

    Call \@gls@link
      \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
    }%

    \glspostlinkhook
  }

\@glstext@
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glsentrytext{#2}}{#2}}#3}%
}

\@Glstext@
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}}#3}%
}

\@GLStext@
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glstextaccessdisplay{\glsuppercase{\glsentrytext{#2}}}{#2}%
  \glsuppercase{#3}}%
}

```

```

\@glsfirst@
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glsentryfirst{#2}}{#2}#3}%
}

\@Glsfirst@
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}

\@GLSfirst@
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\glsupercase{\glsentryfirst{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsplural@
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glsentryplural{#2}}{#2}#3}%
}

\@Glsplural@
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}

\@GLSplural@
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\glsupercase{\glsentryplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsfirstplural@
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glsentryfirstplural{#2}}{#2}#3}%
}

\@Glsfirstplural@
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}

\@GLSfirstplural@
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\glsupercase{\glsentryfirstplural{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsname@
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}

\@Glsname@
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}

\@GLSname@
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsnameaccessdisplay{\glsuppercase{\glsentryname{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdesc@
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}

\@Glsdesc@
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}

\@GLSdesc@
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionaccessdisplay{\glsuppercase{\glsentrydesc{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

\@glsdescplural@
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}

\@Glsdescplural@
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}

\@GLSdescplural@
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsdescriptionpluralaccessdisplay{\glsuppercase{\glsentrydescplural{#2}}}{#2}%
    \glsuppercase{#3}}%
  }
}

```

```

\@glssymbol@
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}

\@Glsymbol@
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}

\@GLSsymbol@
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glssymbolaccessdisplay{\glssupercase{\glsentrysymbol{#2}}{#2}%
      \glssupercase{#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{\glssupercase{\glsentrysymbolplural{#2}}{#2}%
      \glssupercase{#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{\glssupercase{\glsentryuseri{#2}}{#2}%
      \glssupercase{#3}}}%
  }
}

```

```

\@glsuserii@
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}

\@Glsuserii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}

\@GLSuserii@
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\glsupercase{\glsentryuserii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriii@
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}

\@Glsuseriii@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}

\@GLSuseriii@
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\glsupercase{\glsentryuseriii{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuseriv@
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}

\@Glsuseriv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}

\@GLSuseriv@
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\glsupercase{\glsentryuseriv{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

```

\@glsuserv@
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}

\@Glsuserv@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

\@GLSuserv@
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuservaccessdisplay{\glsupercase{\glsentryuserv{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

\@glsuservi@
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}

\@Glsuservi@
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}

\@GLSuservi@
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserviaccessdisplay{\glsupercase{\glsentryuservi{#2}}}{#2}%
    \glsupercase{#3}}%
  }
}

```

6.3 Displaying the Glossary

We need to redefine the way the glossary entries are formatted to include the accessibility support. The predefined glossary styles use `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol`, but we need to provide compatibility with earlier versions in case users have defined their own styles using `\accsuppglossaryentryfield` and `\accsuppglossarysubentryfield`.

Now redefine `\glossentryname`, `\glossentrydesc` and `\glossentrysymbol` etc so they use the accessibility stuff.

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}

```

```

\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}

\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}

```

\accsuppglossaryentryfield

```

\newcommand*{\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}

```

\accsuppglossarysubentryfield

```

\newcommand*{\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}

```

6.4 Acronyms

Redefine acronym styles provided by glossaries:

long-short $\langle long \rangle$ ($\langle short \rangle$) acronym style.

```
\renewacronymstyle{long-short}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1}}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glseyentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```

short-long $\langle short \rangle$ ($\langle long \rangle$) acronym style.

```
\renewacronymstyle{short-long}%
{%
```

Check for long form in case this is a mixed glossary.

```
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glslongaccessdisplay{\glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glslongaccessdisplay{\Glseyentrylong{##1}}{##1}##2\space
\glsshortaccessdisplay
{\protect\firstacronymfont{\glseyentryshort{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glslongpluralaccessdisplay{\Glseyentrylongpl{##1}}{##1}##2\space
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glseyentryshortpl{##1}}{##1}}%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glseyentryshort{##1}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
```



```

\glsshortaccessdisplay
  {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay
    {\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

`long-short-desc` $\langle long \rangle$ ($\{\langle short \rangle\}$) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
  }

```

`long-sc-short-desc` $\langle long \rangle$ ($\text{\textsc{\langle short \rangle}}$) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%

```

```

{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

long-sm-short-desc <long> (\textsmaller{<short>}) acronym style that has an accompanying de-
scription (which the user needs to supply).
\renewacronymstyle{long-sm-short-desc}%
{
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

short-long-desc <short> ({<long>}) acronym style that has an accompanying description (which
the user needs to supply).
\renewacronymstyle{short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

sc-short-long-desc <long> (\textsc{<short>}) acronym style that has an accompanying description
(which the user needs to supply).
\renewacronymstyle{sc-short-long-desc}%
{
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}

```

sm-short-long-desc *<long>* (`\textsmaller{<short>}`) acronym style that has an accompanying description (which the user needs to supply).

```

\renewacronymstyle{sm-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}

```

dua *<long>* only acronym style.

```

\renewacronymstyle{dua}%
{%

```

Check for long form in case this is a mixed glossary.

```

\ifdefempty\glscustomtext
{%
\ifglshaslong{\glslabel}%
{%
\glsifplural
{%

```

Plural form:

```

\glscapscase
{%

```

Plural form, don't adjust case:

```

\glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
\glsinsert
}%
{%

```

Plural form, make first letter upper case:

```

\glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
\glsinsert
}%
{%

```

Plural form, all caps:

```

\glslongpluralaccessdisplay
{\glsupercase{\glsentrylongpl{\glslabel}}}{\glslabel}%

```

```

        \glsuppercase{\glsinsert}%
    }%
}%
{%
Singular form
        \glscapscase
    {%
Singular form, don't adjust case:
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, make first letter upper case:
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
    }%
    {%
Subsequent singular form, all caps:
        \glslongaccessdisplay
        {\glsuppercase
         {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
        \glsuppercase{\glsinsert}%
    }%
}%
{%
Not an acronym:
        \glsgenentryfmt
    }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glslongaccessdisplay
        {\glsuppercase{\glsentrylong{##2}}{##2}##3\space
        (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%

```

```

\glslongpluralaccessdisplay
  {\glsentrylongpl{##2}}{##2}##3\space
(\glsshortpluralaccessdisplay
  {\acronymfont{\glsentryshortpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsupercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
      {\acronymfont{\glsentryshortpl{##2}}{##2}})%
  }%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}{##1}})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}

```

dua-desc *<long>* only acronym style with user-supplied description.

```

\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%

```

}%

footnote *<short>*\footnote{*<long>*} acronym style.

\renewacronymstyle{footnote}%

{%

Check for long form in case this is a mixed glossary.

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%

}%

{%

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%

Need to ensure hyperlinks are switched off on first use:

\glshyperfirstfalse

\renewcommand*{\genacrfullformat}[2]{%

\glsshortaccessdisplay

{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%

\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%

}%

\renewcommand*{\Genacrfullformat}[2]{%

\glsshortaccessdisplay

{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%

\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%

}%

\renewcommand*{\genplacrfullformat}[2]{%

\glsshortpluralaccessdisplay

{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%

\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%

}%

\renewcommand*{\Genplacrfullformat}[2]{%

\glsshortpluralaccessdisplay

{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%

\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%

}%

\renewcommand*{\acronymentry}[1]{%

\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%

\renewcommand*{\acronymsort}[2]{##1}%

\renewcommand*{\acronymfont}[1]{##1}%

\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%

Don't use footnotes for \acrfull:

\renewcommand*{\acrfullfmt}[3]{%

\glslink[##1]{##2}{%

\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space

(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%

\renewcommand*{\Acrfullfmt}[3]{%

\glslink[##1]{##2}{%

\glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space

(\glslongaccessdisplay{\glsentrylong{##2}}{##2}))}%

\renewcommand*{\ACRfullfmt}[3]{%

\glslink[##1]{##2}{%

```

\glsshortaccessdisplay
  {\glssupercase
    {\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\glslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
      (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\glssupercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\glslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%

```

Similarly for \glentryfull etc:

```

\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1}))%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1}))%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}))%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
    (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}))%
}

```

footnote-sc \textsc{<short>}\footnote{<long>} acronym style.

```

\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%

```

footnote-sm \textsmaller{<short>}\footnote{<long>} acronym style.

```
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
```

footnote-desc <short>\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sc-desc \textsc{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1))}%
  }
}
```

footnote-sm-desc \textsmaller{<short>}\footnote{<long>} acronym style that has an accompanying description (which the user needs to supply).

```
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}%
```



```
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}
```

```
\glsglsdefaultshortaccess{\<long>}{\<short>}
```

`\glsglsdefaultshortaccess`

Default shortaccess value.

```
\newcommand*{\glsglsdefaultshortaccess}[2]{#1}
```

Use `\newacronymhook` to modify the key list to set the access text to the long version by default.

```
\renewcommand*{\newacronymhook}{%
\protected@edef\@gls@keylist{%
shortaccess={\glsglsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
shortpluralaccess={\glsglsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
\the\glskeylisttok}%
\expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
```

`\DefaultNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionFootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\DescriptionNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\FootnoteNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

`\SmallNewAcronymDef` Deprecated with the introduction of `\setacronymstyle` in v4.02 (2013-12-05) and removed in v4.50.

The following are kept for compatibility with versions before 3.0:

`\glsshortaccesskey`

```
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
```

`\glsshortpluralaccesskey`

```
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
```

`\glslongaccesskey`

```
\newcommand*{\glslongaccesskey}{\glslongkey access}%
```

```
\glslongpluralaccesskey
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
```

6.5 Debugging Commands

```
\showglongnameaccess
\newcommand*{\showglongnameaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
```

```
\showglongtextaccess
\newcommand*{\showglongtextaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
```

```
\showglongpluralaccess
\newcommand*{\showglongpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
```

```
\showglongfirstaccess
\newcommand*{\showglongfirstaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
```

```
\showglongfirstpluralaccess
\newcommand*{\showglongfirstpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
```

```
\showglongsymbolaccess
\newcommand*{\showglongsymbolaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
```

```
\showglongsymbolpluralaccess
\newcommand*{\showglongsymbolpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}
```

```
\showglongdescaccess
\newcommand*{\showglongdescaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
```

```
\showglongdescpluralaccess
\newcommand*{\showglongdescpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
```

```

\showgloshortaccess
\newcommand*{\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@shortaccess\endcsname
}

\showgloshortpluralaccess
\newcommand*{\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@shortpluralaccess\endcsname
}

\showglolongaccess
\newcommand*{\showglolongaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@longaccess\endcsname
}

\showglolongpluralaccess
\newcommand*{\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@glstetoklabel{#1}@longpluralaccess\endcsname
}

```

7 Backwards Compatibility

7.1 glossaries-compatible-207

Provides compatibility with version 2.07 and below. This uses original `glossaries` xindy and `makeindex` formatting, so can be used with old documents that had customized style files, but hyperlinks may not work properly.

Now only available with rollback.

```
\NeedsTeXFormat{LaTeX2e}
```

Rollback releases:

```

\DeclareRelease{v4.46}{2020-03-19}{glossaries-compatible-207-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-compatible-207-2021-11-01.sty}
\DeclareCurrentRelease{v5.1}{2026-04-21}

```

Identify package.

```
\ProvidesPackage{glossaries-compatible-207}[2026/04/21 v5.1 (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{v5.1}{2026-04-21}
```

Identify package.

```
\ProvidesPackage{glossaries-compatible-307}[2026/04/21 v5.1 (NLCT)]
```

Trigger error and that's it.

```
\PackageError{glossaries}  
{glossaries-compatible-307.sty no longer available. Use rollback if you need it}%  
{Try \string\usepackage{glossaries}[=v4.46]}
```

8 Rollback

8.1 Rollback v4.46 (glossaries-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}  
\ProvidesPackage{glossaries}[2020/03/19 v4.46 (NLCT)]  
\RequirePackage{ifthen}  
\RequirePackage{xkeyval}[2006/11/18]  
\RequirePackage{mfirstuc}  
\RequirePackage{textcase}  
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%  
\RequirePackage{xfor}  
\RequirePackage{datatool-base}  
\RequirePackage{amsgen}  
\RequirePackage{etoolbox}  
\newif\if@gls@docloaded  
\@ifpackageloaded{doc}%  
{%  
  \@gls@docloadedtrue  
}%  
{%  
  \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%  
}  
\if@gls@docloaded  
  \let\glsorg@theglossary\theglossary  
  \let\glsorg@endtheglossary\endtheglossary  
  \let\glsorg@PrintChanges\PrintChanges  
  \renewcommand{\PrintChanges}{%  
    \begingroup  
      \let\theglossary\glsorg@theglossary  
      \let\endtheglossary\glsorg@endtheglossary  
      \glsorg@PrintChanges  
    \endgroup  
  }  
\fi
```

```

\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
\ifcase\gls@debug@nr\relax
% debug=true
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\let\@glsshowtarget\@gobble
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand*\@glsshowtarget{\@@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand*\@glsshowaccsupp{\@glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\nfss@text{\glsshowtargetfont [#1]}%
\else
\ifinner
{\glsshowtargetfont [#1]}%

```

```

\else
\glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetouter}[1]{%
\glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbols{{\tiny$\triangleright$}}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*{\@glsshowtarget}[1]{%
\begingroup
\protected@edef\@gls@tmp{#1}%
\@onelevel@sanitize\@gls@tmp
\expandafter\glsshowtarget\expandafter{\@gls@tmp}%
\endgroup
}
\newcommand*\@glsshowaccsupp}[3]{%
\newrobustcmd*{\glsshowaccsupp}[3]{%
\ifstrempy{#1}%
{\@glsshowtarget{/#2 (#3)}}%
{\@glsshowtarget{/#2 (#3) [#1]}}%
}
}
\newcommand*\@gls@see@noindex{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries\space (or move
\string\newglossaryentry\space
definitions into the preamble)}}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key. It may
be that the 'see' key has been written to the .glsdefs
file from the previous run, in which case you need to
move your definitions
to the preamble if you don't want to use
\string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenindex}%
[\gls@seenindex@val\gls@seenindex@nr]{error,warn,ignore}{%
\ifcase\gls@seenindex@nr
\renewcommand*\@gls@see@noindex{%
\PackageError{glossaries}%
{'\gls@xr@key' key may only be used after \string\makeglossaries\space
or \string\makenoidxglossaries}%
{You must use \string\makeglossaries\space
or \string\makenoidxglossaries\space before defining
any entries that have a '\gls@xr@key' key}%
}%
}

```

```

\or
\renewcommand*{\@gls@see@noindex}{%
  \GlossariesWarning{'\gls@xr@key' key ignored}%
}%
\or
\renewcommand*{\@gls@see@noindex}{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{}%
\renewcommand*{\@glossarysec}{#1}%
\newcommand*{\@glossarysecstar}{*}%
\newcommand*{\@glossaryseclabel}{}%
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{}%
false,nolabel,autolabel,nameref}[nolabel]{}%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{}%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}

```

```

\@gls@declareoption{nonumberlist}{%
  \renewcommand*{\glossaryentrynumbers}[1]{\@gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\glssavenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*{\@glo@seeautonumberlist}{%
    \def\@glo@prefix{\glsnextpages}%
  }%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\glseclocationstrue
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}[=v4.46]}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}%
\IfFileExists{supertabular.sty}{%
  \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}[=v4.46]}{}%
  \newcommand*\@gls@loadsuper{}%
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}%
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}[=v4.46]}%
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
    \ifdefstring{\@glossary@default@style}{list}%
    {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}[=v4.46]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}%
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*\@gls@postdescription{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}

```



```

}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglentrycounter
    \ifundef\c@glossaryentry
      {%
        \ifx\@gls@counterwithin\@empty
          \newcounter{glossaryentry}%
        \else
          \newcounter{glossaryentry}[\@gls@counterwithin]%
        \fi
        \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
      }%
    }%
  \fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
    {%
      \ifglssubentrycounter
        \ifglentrycounter
          \newcounter{glossarysubentry}[glossaryentry]%
        \else
          \newcounter{glossarysubentry}%
        \fi
        \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
      \fi
    }%
  }%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \ifx\@glo@sort\@glsdefaultsort
    \let\@glo@sort\@glo@name
  \fi
  \let\glsdosanitizesort\@gls@sanitizesort
  \glsprestandardsort{\@glo@sort}{##1}{##2}%
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glsortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}
\newcommand*\@gls@setupsort@def{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \ifcsundef{glossary@##1@sortcount}%
  {\@gls@defsortcount{##1}}%
  {}%
  \expandafter\global\expandafter
  \advance\csname glossary@##1@sortcount\endcsname by 1\relax
  \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
    \expandafter\glsortnumberfmt
    {\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
  \expandafter\global
  \expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
  \expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%

```

```

\edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\edef\@glo@type{\csname glo@##1@type\endcsname}%
\edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected\xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[gls2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
\ifglsacronym
\renewcommand{\@gls@do@acronymsdef}{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%

```

```

\renewcommand*{\acronymtype}{acronym}%
\newcommand*{\gls@tr@set@acronym@toctitle}{%
  \translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
\else
\let\@gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
\ifglsacronym
\ifbool{glscompatible-3.07}%
{}%
{%
\providecommand*{\printacronyms}[1][{}]{%
\printglossary[type=\acronymtype,#1]}%
}%
\fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
\glsacronymtrue
\def\@gls@do@acronymsdef{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*{\acronymtype}{acronym}%
\newcommand*{\gls@tr@set@acronym@toctitle}{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
}
\newcommand*{\@glsacronymlists}{}
\newcommand*{\@addtoacronymlists}[1]{%
\ifx\@glsacronymlists\@empty
\protected@xdef\@glsacronymlists{#1}%
\else
\protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
\fi
}
\newcommand*{\DeclareAcronymList}[1]{%
\glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
\edef\@do@gls@islistofacronyms{%
\noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
\@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
\def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
\def\@gls@before{##1}\def\@gls@after{##2}}%

```

```

\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
  #4%
\else
  #3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@gls@acronymlists}{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
  \fi
}

```

```

\renewcommand*\GlossariesWarningNoLine}[1]{}%
\renewcommand*\glosortentrieswarning{}%
\renewcommand*\@gls@missinglang@warn}[2]{}%
\fi
}
\newcommand*\@gls@missinglang@warn}[2]{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘#1’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-#2’ or similar}%
}
\@gls@declareoption{nolangwarn}{%
\renewcommand*\@gls@missinglang@warn}[2]{}%
}
\newcommand*\@gls@warnonglossdefined{%
\GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*\@gls@warnontheGLOSSdefined{%
\GlossariesWarning{Overriding ‘theGLOSS’ environment}%
}
\@gls@declareoption{noredefwarn}{%
\renewcommand*\@gls@warnonglossdefined{}%
\renewcommand*\@gls@warnontheGLOSSdefined{}%
}
\newcommand*\@gls@sanitizedesc{%
}
\newcommand*\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename{}
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort{%
\ifgls@sanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort

```

```

\fi
}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*{\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let\'\@firstofone
  \let\'\@firstofone
  \let\^@\@firstofone
  \let\@"\@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let\=\@firstofone
  \let\.\@firstofone

```

```

\let~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glSsetnoexpandfield{desc}%
\glSsetnoexpandfield{descplural}%
\else
\glSsetexpandfield{desc}%
\glSsetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glSsetnoexpandfield{name}%
\else
\glSsetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glSsetnoexpandfield{symbol}%

```



```

        \glsssetnoexpandfield{symbolplural}%
    \else
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
    \fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }{%
        \setkeys[gls]{sanitize}{#1}%
    }%
}

```

```

}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
  {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}%
[\gls@translate@val\gls@translate@nr]%
{true,false,babel}[true]%
{%
  \ifcase\gls@translate@nr\relax
  \glstranslatetrue
  \renewcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \@ifpackageloaded{babel}%
      {%
        \IfFileExists{translator.sty}%
        {%
          \RequirePackage{translator}%
          \let\glsifusetranslator\@firstoftwo
        }%
      }%
    }%
  }%
}

```

```

        }%
        {}%
    }%
}%
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\endfortrue
}%
}%
}
}
\define@boolkey{glossaries.sty}[gls]{indexonlyfirst}[true]{}
\glsindexonlyfirstfalse
\define@boolkey{glossaries.sty}[gls]{hyperfirst}[true]{}
\glshyperfirsttrue
\newcommand*{\@gls@setacrstyle}{}
\define@boolkey{glossaries.sty}[glsacr]{footnote}[true]{}
\ifbool{glsacrdescription}%
{%
}%
\renewcommand*{\@gls@sanitizedesc}{}%
}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{description}[true]{}
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smallcaps}[true]{}

```

```

\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{smaller}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{dua}[true]{%
\renewcommand*{\@gls@sanitizesymbol}{}%
\renewcommand*{\@gls@setacrstyle}{\SetAcronymStyle}%
}
\define@boolkey{glossaries.sty}[glsacr]{shortcuts}[true]{%
\newcommand*{\glsorder}{word}
\newcommand*{\@glsorder}[1]{%
\define@choicekey{glossaries.sty}{order}{word,letter}{%
\def\glsorder{#1}}
\newif\ifglsxindy
\glsxindyfalse
\@gls@declareoption{makeindex}{\glsxindyfalse}
\define@boolkey[gls]{xindy}{glsnumbers}[true]{%
\glsxindy@glsnumberstrue
\def\@xdy@main@language{\language}%
\define@key[gls]{xindy}{language}{\def\@xdy@main@language{#1}}
\ifcsundef{inputencodingname}{%
\def\gls@codepage{}{%
\def\gls@codepage{\inputencodingname}
}
\define@key[gls]{xindy}{codepage}{\def\gls@codepage{#1}}
\define@key{glossaries.sty}{xindy}[]{%
\glsxindytrue
\setkeys[gls]{xindy}{#1}%
}
\@gls@declareoption{xindygloss}{%
\glsxindytrue
}
\@gls@declareoption{xindynoglsnumbers}{%
\glsxindytrue
\glsxindy@glsnumbersfalse
}
\providecommand{\@domakeglossaries}[1]{#1}
\@gls@declareoption{disablemakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect
(\string\makenoidxglossaries\space has already been used)}}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option ‘disablemakegloss’ has no effect

```

```

        (\string\makeglossaries\space has already been used))}%
    }%
    {%
        \renewcommand{\@domakeglossaries}[1]{%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been disabled}%
        }%
    }%
}%
}
\@gls@declareoption{restoremakegloss}{%
    \ifdefequal\makeglossaries\@no@makeglossaries
    {%
        \GlossariesWarning{Option 'restoremakegloss' has no effect
        (\string\makenoidxglossaries\space has already been used))}%
    }%
    {%
        \ifdefequal\@makeglossary\@gobble
        {%
            \GlossariesWarning{Option 'restoremakegloss' has no effect
            (\string\makeglossaries\space has already been used)}%
        }%
        {%
            \PackageInfo{glossaries}{\string\makeglossaries\space and
            \string\makenoidxglossaries\space have been restored}%
            \let\@domakeglossaries\@firstofone
        }%
    }%
}%
}
\newcommand*{\@do@write@glslabels}{%
    \AtEndDocument{\@do@write@glslabels}%
    \let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
    \newwrite\@gls@labelsfile
    \immediate\openout\@gls@labelsfile=\jobname.glslabels
    \forallglsentries[\@gls@types,\@ignored@glossaries]{\@glssentry}%
    {\ifdefempty{\@glssentry}{\immediate\write\@gls@labelsfile{\@glssentry}}}%
    \immediate\closeout\@gls@labelsfile
}
\@gls@declareoption{writeglslabels}{\@do@write@glslabels}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
    [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
    \ifnum\gls@automake@nr=1\relax
        \glsautomakefalse
    \else
        \glsautomaketrue
    \fi
}

```

```

\ifglsautomake
  \renewcommand*{\@gls@doautomake}{%
    \PackageError{glossaries}{You must use
    \string\makeglossaries\space with automake=true}%
    {%
      Either remove the automake=true setting or
      add \string\makeglossaries\space to your document preamble.%
    }%
  }%
\else
  \renewcommand*{\@gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifglssavewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
instead. If you \MessageBreak need the original definition of
\string\makeglossary\space use \MessageBreak the package options
kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict)}}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed

```

```

with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict))}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict))}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][\@gls@tr@set@symbols@toctitle]{%
\printglossary[type=symbols,##1]}%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef

```

```

}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\]{\printglossary[type=index,##1]}%
  \newcommand*{\newterm}[2][\]{%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \expandtwoargs
    \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@option
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.46]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts

```



```

        \DefineAcronymSynonyms
    \fi
}%
\fi
\glsacrshortcutsfalse
\let\@gls@do@numbersdef\relax
\let\@gls@do@symbolssdef\relax
\let\@gls@do@indexdef\relax
\let\@gls@do@acronymsdef\relax
\ifgl Sentrycounter
    \let\@gls@doentrycounterdef\relax
\else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
\fi
\ifgl subentrycounter
    \let\@gls@dosubentrycounterdef\relax
\else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
\fi
\setkeys{glossaries.sty}{#1}%
\@gls@setacrstyle
\@gls@setupshortcuts
\@gls@do@acronymsdef
\@gls@do@numbersdef
\@gls@do@symbolssdef
\@gls@do@indexdef
\@gls@doentrycounterdef
\@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
{%
    \ifcsundef{chapter}{}%
    {%
        \let\@gls@old@chapter\@chapter
        \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
        \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}{}}%
    }%
}%
{}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
    \ifx\@gls@onlypremakeg\@empty
        \def\@gls@onlypremakeg{#1}%
    \else
        \expandafter\toks@\expandafter{\@gls@onlypremakeg}%
        \edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
    \fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%

```

```

\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
  \def#1{\PackageError{glossaries}{\string#1\space may only be
    used before \string\makeglossaries}{You can't use
    \string#1\space after \string\makeglossaries}}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
  \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}
\providecommand*{\glsymbolsgroupname}{Symbols}
\providecommand*{\glsnumbersgroupname}{Numbers}
\newcommand*{\glspluralsuffix}{s}
\newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*{\glsupacrpluralsuffix}{\glsup{t}\glsacrpluralsuffix}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
  \ifundefined{ver@glossaries-#1.1df}{\input{glossaries-#1.1df}}{}}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
  \ProvidesFile{glossaries-#1.1df}%
}
\newcommand*{\addglossarytocaptions}[1]{}
\ifglstranslate
  \RequirePackage{tracklang}
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*{\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp

```

```

        \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
}%
}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
  }-}%
\fi
\DeclareRobustCommand*{\nopostdesc}{%
\newcommand*{\@nopostdesc}{%

```

```

\let\org@glspostdescription\glspostdescription
\def\glspostdescription{%
  \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gl@istfilebase}{#1}%
  \ifglxsindy
    \def\istfilename{\gl@istfilebase.xdy}
  \else
    \def\istfilename{\gl@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglxsindy
  \def\istfilename{\gl@istfilebase.xdy}
\else
  \def\istfilename{\gl@istfilebase.ist}
\fi
\newcommand*{\gl@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{%
\newcommand*{\glscpositor}{.}
\newcommand*{\glSetCompositor}[1]{%
  \renewcommand*{\glscpositor}{#1}}
\@onlypremakeg\glSetCompositor
\newcommand*{\@gl@Alphacompositor}{\glscpositor}
\ifglxsindy
  \newcommand*{\glSetAlphaCompositor}[1]{%
    \renewcommand*{\@gl@Alphacompositor}{#1}}
\else
  \newcommand*{\glSetAlphaCompositor}[1]{%
    \gl@noxindywarning\glSetAlphaCompositor}
\fi
\@onlypremakeg\glSetAlphaCompositor
\newcommand*{\gl@suffixF}{%
\newcommand*{\glSetSuffixF}[1]{%
  \renewcommand*{\gl@suffixF}{#1}}
\@onlypremakeg\glSetSuffixF
\newcommand*{\gl@suffixFF}{%
\newcommand*{\glSetSuffixFF}[1]{%
  \renewcommand*{\gl@suffixFF}{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\gl@numberformat}[1]{#1}%
}%
{%
  \newcommand*{\gl@numberformat}[1]{\glshypernumber{#1}}%
}

```

```

\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else
        \@mkboth{#1}{#1}%
      \fi
    }
  }
}
\providecommand{\glossarymark}[1]{%
  \ifglsucmark
    \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%

```

```

\else
  \@mkboth{#1}{#1}%
\fi
}
\newcommand*\setglossarysection[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*\@glossarysection[2]{%
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname[#1]{#2}%
  }%
  {%
    \csname\@glossarysec\endcsname*{#2}%
    \gls@toc{#1}{\@glossarysec}%
  }%
  \@glossaryseclabel
}
\newcommand*\@pglossarysection[2]{%
  \glsclearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%
    \gls@toc{#1}{\@glossarysec}%
    \csname\@glossarysec\endcsname*{#2}%
  }%
  \@glossaryseclabel
}
\newcommand*\gls@docclearpage{%
  \ifthenelse{\equal{\@glossarysec}{chapter}}{%
    {%
      \ifcsundef{cleardoublepage}%
      {%
        \clearpage
      }%
      {%
        \ifcsdef{if@openright}%
        {%
          \if@openright
            \cleardoublepage
          \else
            \clearpage
          \fi
        }%
        {%
          \cleardoublepage
        }%
      }%
    }%
  }%

```

```

}%
{}%
}
\newcommand*{\glsclearpage}{\gls@doclearpage}
\newcommand*{\@gls@toc}[2]{%
  \ifglstoc
    \ifglsnumberline
      \addcontentsline{toc}{#2}{\protect\numberline{#1}}%
    \else
      \addcontentsline{toc}{#2}{#1}%
    \fi
  \fi
}
\newcommand*{\glsnoxindywarning}[1]{%
  \GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*{\glsnomakeindexwarning}[1]{%
  \GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
  \edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
  \edef\xdyattributelist{}%
\fi
\ifglsxindy
  \def\xdylocref{}
\fi
\newcommand*{\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,#1,##2\end@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
  \newcommand*{\@xdycounters}{\glscounter}
  \newcommand*{\GlsAddXdyCounters}[1]{%
    \@for\@gls@ctr:=#1\do{%
      \edef\@do@addcounter{%
        \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
        {%
          \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
            \noexpand\@gls@ctr}%
        }%
      }%
    }%
  }%

```

```

        \@do@addcounter
    }
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
    \glsnxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glssaddxdycounters{%
    \PackageError{glossaries}{\string\GlsAddXdyCounters\space
can't be used after \string\GlsAddXdyAttribute}{Move all
occurrences of \string\GlsAddXdyCounters\space before the first
instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glssaddxdyattribute[2]{%
    \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
\string"#2#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J%
(markup-locref
:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string")}%
    \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
    \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
    \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
    \fi
    \@for\@this@counter:=\@xdycounters\do{%
        \protected@edef\gls@do@addxdyattribute{%
            \noexpand\@glssaddxdyattribute{#1}{\@this@counter}%
        }
        \gls@do@addxdyattribute
    }%
    \let\GlsAddXdyCounters\@disabled@glssaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
    \glsnxindywarning\GlsAddXdyAttribute}
\fi

```



```

\ifglxindy
\newcommand*{\@gls@addpredefinedattributes}{%
  \GlsAddXdyAttribute{glsnumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{glshypernumber}
  \GlsAddXdyAttribute{hyperrm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{glsignore}
}
\else
  \let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifglxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ^~J
      (define-alphabet "#1" (#2))}%
  }
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxindy
  \newcommand*{\@gls@xdy@locationlist}{%
    roman-page-numbers,%
    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
  }
\protected@edef\@gls@roman{\@roman{0}\string"

```

```

\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{%
\string"roman-numbers-lowercase\string"%
}%
\else
\expandafter
\edef\csname @gls@xdy@Lclass@roman-page-numbers\endcsname{
:sep \string"\@gls@roman\string"%
}%
\fi
\expandafter\def\csname @gls@xdy@Lclass@Roman-page-numbers\endcsname{%
\string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@arabic-page-numbers\endcsname{%
\string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@alpha-page-numbers\endcsname{%
\string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Alpha-page-numbers\endcsname{%
\string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy@Lclass@Appendix-page-numbers\endcsname{%
\string"ALPHA\string"
:sep \string"\@gls@Alphacompositor\string"
\string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy@Lclass@arabic-section-numbers\endcsname{%
\string"arabic-numbers\string"
:sep \string"\glscompositor\string"
\string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxindy
\newcommand*\@GlsAddXdyLocation}[3][[]]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^~J%

```

```

        (define-location-class \string"#2\string"^^J\space\space
        \space(:sep \string"{}\glspenbrace\string" #3
        :sep \string"\glsclosebrace\string"))
    }%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glspenbrace"
#1
:sep "\glsclosebrace\glspenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnnoxindywarning\GlsAddXdyLocation}
\fi
\ifglxindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnnoxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
\def\@xdysortrules{}
\fi
\ifglxindy
\newcommand*\GlsAddSortRule[2]{%
\expandafter\toks@\expandafter{\@xdysortrules}%
\protected@edef\@xdysortrules{\the\toks@ ^^J

```

```

        (sort-rule \string"#1\string" \string"#2\string"))}%
    }
\else
    \newcommand*\GlsAddSortRule[2]{%
        \glsnoxywarning\GlsAddSortRule}
\fi
\ifglxindy
    \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
    \newcommand*\GlsAddXdyStyle[1]{%
        \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
    \newcommand*\GlsAddXdyStyle[1]{%
        \glsnoxywarning\GlsAddXdyStyle}
\fi
\ifglxindy
    \newcommand*\GlsSetXdyStyles[1]{%
        \edef\@xdyrequiredstyles{#1}}
\else
    \newcommand*\GlsSetXdyStyles[1]{%
        \glsnoxywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{}
\def\@xdylanguage#1#2{}
\ifglxindy
    \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
        \ifglossaryexists{#1}{%
            \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
        }{%
            \PackageError{glossaries}{Can't set language type for
                glossary type '#1' --- no such glossary}{%
                You have specified a glossary type that doesn't exist}}
\else
    \newcommand*\GlsSetXdyLanguage[2][{}]{%
        \glsnoxywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
    \newcommand*\GlsSetXdyCodePage[1]{%
        \renewcommand*\@gls@codepage{#1}%
    }
    \AtBeginDocument{%
        \ifx\@gls@codepage\@empty
            \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
        \fi
    }
\else
    \newcommand*\GlsSetXdyCodePage[1]{%
        \glsnoxywarning\GlsSetXdyCodePage}

```

```

\fi
\ifglxsindy
\ifglxsindy@glslnumbers
\def\@xdylettergroups{(define-letter-group
\string"glslnumbers\string"^^J\space\space\space
:prefixes (\string"0\string" \string"1\string"
\string"2\string" \string"3\string" \string"4\string"
\string"5\string" \string"6\string" \string"7\string"
\string"8\string" \string"9\string")^^J\space\space\space
\@xdynumbergrouporder)}
\else
\def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
\expandafter\toks@\expandafter{\@xdylettergroups}%
\protected@edef\@xdylettergroups{\the\toks@^^J%
(define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
\@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
\@for#1:=\@glslacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forallglseentries[3][\@glsldefaulttype]{%
\edef\@@glo@list{\csname glolist@#1\endcsname}%
\@for#2:=\@@glo@list\do
{%
\ifdefempty{#2}{\fi}%
}%
}
\newcommand*\forallglseentries[3][\@glo@types]{%
\expandafter\forallglossaries\expandafter[#1]{\@@this@glo@}%
{%
\forallglseentries[\@@this@glo@]{#2}{#3}%
}%
}
\newcommand*\ifglossaryexists{%
\@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand*\@ifglossaryexists[3]{%
\ifcsundef{\@glo@type@#1@out}{#3}{#2}%
}
\newcommand*\s@ifglossaryexists[3]{%
\ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glslsetoklabel[1]{#1}
\newcommand*\ifglseentryexists[3]{%

```

```

\ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@glseentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs\glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
        {%

```

```

        \def\do@glshaschildren{#2}%
        \@endfortrue
    }%
    {}%
}%
\do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
    \glstoifexists{#1}%
    {%
        \ifcsepty{glo@\glstetoklabel{#1}@parent}{#3}{#2}%
    }%
}
\newcommand*{\ifglshasdesc}[3]{%
    \ifcsepty{glo@\glstetoklabel{#1}@desc}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
    \ifcsequal{glo@\glstetoklabel{#1}@desc}{@no@post@desc}%
    {#2}%
    {#3}%
}
}
\newrobustcmd*{\ifglshassymbol}[3]{%
    \letcs{\@glo@symbol}{glo@\glstetoklabel{#1}@symbol}%
    \ifdefempty\@glo@symbol
    {#3}%
    {%
        \ifdefequal\@glo@symbol\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshaslong}[3]{%
    \letcs{\@glo@long}{glo@\glstetoklabel{#1}@long}%
    \ifdefempty\@glo@long
    {#3}%
    {%
        \ifdefequal\@glo@long\@gls@default@value
        {#3}%
        {#2}%
    }%
}
}
\newrobustcmd*{\ifglshasshort}[3]{%
    \letcs{\@glo@short}{glo@\glstetoklabel{#1}@short}%
    \ifdefempty\@glo@short
    {#3}%
    {%
        \ifdefequal\@glo@short\@gls@default@value

```

```

        {#3}%
        {#2}%
    }%
}
\newrobustcmd*{\ifglshasfield}[4]{%
    \glstoifexists{#2}%
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@#1}%
        \ifdef\@glo@thisvalue
        {%
            \ifdefempty\@glo@thisvalue
            {%
                #4%
            }%
        }%
        {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
                #4%
            }%
            {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
            }%
        }%
    }%
}%
{%
    \@gls@fetchfield{\@gls@thisfield}{#1}%
    \ifdef\@gls@thisfield
    {%
        \letcs{\@glo@thisvalue}{glo@glstetoklabel{#2}@ \@gls@thisfield}%
        \ifdefempty\@glo@thisvalue
        {%
            #4%
        }%
        {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
                #4%
            }%
            {%
                \let\glscurrentfieldvalue\@glo@thisvalue
                #3%
            }%
        }%
    }%
}%
{%
    \GlossariesWarning{Unknown entry field '#1'}%
    #4%
}%
}%

```



```

}%
}
\newcommand*{\glscurrentfieldvalue}{%
\newcommand*{\@glo@types}{,}%
\newcommand*{\@gls@provide@newglossary{%
  \protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
  \let\@gls@provide@newglossary\relax
}%
\newcommand*{\defglsentryfmt}[2][\glsdefaultttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}%
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtexext}[1]{%
  \ifboolexpr{test \ifdefstring{#1}{tex}}
    or test \ifdefstring{#1}{TEX}}
  {%
    \def#1{nottex}%
    \PackageError{glossaries}%
      {Forbidden '.tex' extension replaced with '.nottex'}%
      {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
      Don't use '.tex' as an extension for a temporary file.}%
  }%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[\@gls@gobbleopt]}
\def\@gls@gobbleopt[#1]{%
\newcommand*{\newglossary}{\@ifstar\s@newglossary\@ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}%
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaultttype
    {%
      \gdef\glsdefaultttype{#2}%
    }%
  }%
  \toks@{#2}\edef\@glo@types{\@glo@types\the\toks@,}%
  \expandafter\gdef\csname glolist@#2\endcsname{,}%
  \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
  \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
  \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
  \expandafter\@gls@forbidtexext\csname @glotype@#2@log\endcsname
  \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
  \expandafter\@gls@forbidtexext\csname @glotype@#2@out\endcsname
  \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
  \@gls@provide@newglossary
  \protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}%
  \ifcsundef{gls@#2@entryfmt}%

```

```

{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglsxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \edef\@ignored@glossaries{#1}%
  }%
  {%
    \eappto\@ignored@glossaries{, #1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%

```

```

    {%
      \eappto\@gls@nohyperlist{, #1}%
    }%
  }
  \newcommand*{\@ignored@glossaries}{}
  \newcommand*{\ifignoredglossary}[3]{%
    \edef\@gls@igtype{#1}%
    \expandafter\DTLifinlist\expandafter
      {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
  }
  \define@key{glossentry}{name}{%
    \def\@glo@name{#1}%
  }
  \define@key{glossentry}{description}{%
    \def\@glo@desc{#1}%
  }
  \define@key{glossentry}{descriptionplural}{%
    \def\@glo@descplural{#1}%
  }
  \define@key{glossentry}{sort}{%
    \def\@glo@sort{#1}}
  \define@key{glossentry}{text}{%
    \def\@glo@text{#1}%
  }
  \define@key{glossentry}{plural}{%
    \def\@glo@plural{#1}%
  }
  \define@key{glossentry}{first}{%
    \def\@glo@first{#1}%
  }
  \define@key{glossentry}{firstplural}{%
    \def\@glo@firstplural{#1}%
  }
  \newcommand*{\@gls@default@value}{\relax}
  \define@key{glossentry}{symbol}{%
    \def\@glo@symbol{#1}%
  }
  \define@key{glossentry}{symbolplural}{%
    \def\@glo@symbolplural{#1}%
  }
  \define@key{glossentry}{type}{%
    \def\@glo@type{#1}}
  \define@key{glossentry}{counter}{%
    \ifcsundef{c@#1}%
    {%
      \PackageError{glossaries}%
        {There is no counter called ‘#1’}%
      {%
        The counter key should have the name of a valid counter
        as its value%
      }
    }
  }

```

```

    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning{glossaries}%
  {'\gls@xr@key' key doesn't have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
[\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*{\@gls@savenonumberlist}[1]{}
\newcommand*{\@gls@initnonumberlist}{}%
\newcommand*{\@gls@storenonumberlist}[1]{}
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%

```

```

\ifdef\@glo@nonumberlist
{%
  \cslet{glo@glsetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
}%
{}%
}%
\appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@glsnoname}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You
  haven't specified the entry name}}
\newcommand*{\@glsnodelsc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}

```

```

    }%
}%
\newcommand*{\@glsdefaultplural}{%
\newcommand*{\@gls@missingnumberlist}[1]{%
??%
\ifglssavenumberlist
\GlossariesWarning{Missing number list for entry ‘#1’.
Maybe makeglossaries + rerun required}%
\else
\PackageError{glossaries}%
{Package option ‘savenumberlist=true’ required}%
{%
You must use the ‘savenumberlist’ package option
to reference location lists.%
}%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
\let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
\@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}

```

```

\newcommand{\@gls@expand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
      {%
        \@gls@expand@field{#2}{#3}{#4}%
      }%
      {%
        \csuse{gls@assign@#3@field}{#2}{#4}%
      }%
    }%
  }%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\gls@defdocnewglossaryentry}{%
  \let\gls@checkseeallowed\gls@checkseeallowed@preambleonly
  \let\newglossaryentry\new@glossaryentry
}

```

```

\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}
\newrobustcmd{\new@glossaryentry}[2]{%
  \ifundef\@gls@deffile
  {%
    \global\newwrite\@gls@deffile
    \immediate\openout\@gls@deffile=\jobname.glsdefs
  }%
  {}%
  \ifglentryexists{#1}{}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
  \@gls@writedef{#1}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*{\gls@begindocdefs}{%
  \@gls@enablesavenonumberlist
  \edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
  \makeatletter
  \InputIfFileExists{\jobname.glsdefs}{\relax}{\relax}%
  \@gls@restoreat
  \undef\@gls@restoreat
  \gls@defdocnewglossaryentry
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\ifglentryexists{#1}{}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar^^J%
    \string\gls@defglossaryentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{glo@\glsdetoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
      }%
    }%
  }%
}

```



```

        \glspercentchar
    }%
}%
{}%
}%
\glswritedefhook
\immediate\write\@gls@deffile
{%
    \glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar^^J%
    \expandafter\@gobble\string\}\glspercentchar%
}%
}
\newcommand*{\@gls@keymap}{%
    {name}{name},%
    {sort}{sortvalue},% unescaped sort value
    {type}{type},%
    {first}{first},%
    {firstplural}{firstpl},%
    {text}{text},%
    {plural}{plural},%
    {description}{desc},%
    {descriptionplural}{descplural},%
    {symbol}{symbol},%
    {symbolplural}{symbolplural},%
    {user1}{useri},%
    {user2}{userii},%
    {user3}{useriii},%
    {user4}{useriv},%
    {user5}{userv},%
    {user6}{uservi},%
    {long}{long},%
    {longplural}{longpl},%
    {short}{short},%
    {shortplural}{shortpl},%
    {counter}{counter},%
    {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
    \edef\@gls@thisval{#2}%
    \@for\@gls@map:=\@gls@keymap\do{%
        \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
        \ifdefequal{\@this@key}{\@gls@thisval}%
        {%
            \edef#1{\expandafter\@secondoftwo\@gls@map}%
            \@endfortrue
        }%
    }%
}%
}

```

```

\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
}%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
  }%
}

```

```

\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#5' as helper command
   '\expandafter\string\csname @gls@user@#1@\endcsname' already exists}%
  {}%
}%
{%
  \expandafter\newcommand\expandafter*\expandafter
  {\csname @gls@user@#1@\endcsname}[2][\{%
    \new@ifnextchar[%
      {\csuse{@gls@user@#1@}{##1}{##2}}%
      {\csuse{@gls@user@#1@}{##1}{##2}[]}%
    \csdef{@gls@user@#1@}##1##2[##3]{%
      \@gls@field@link{##1}{##2}{#3{##2}##3}%
    }%
    \newrobustcmd*{#5}{%
      \expandafter\@gls@hyp@opt\csname @gls@user@#1@\endcsname}%
    }%
  \ifcsdef{@Gls@user@#1@}%
  {%
    \PackageError{glossaries}%
    {Can't define '\string#6' as helper command
     '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
    {}%
  }%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname @Gls@user@#1@\endcsname}[2][\{%
      \new@ifnextchar[%
        {\csuse{@Gls@user@#1@}{##1}{##2}}%
        {\csuse{@Gls@user@#1@}{##1}{##2}[]}%
      \csdef{@Gls@user@#1@}##1##2[##3]{%
        \@gls@field@link{##1}{##2}{#4{##2}##3}%
      }%
      \newrobustcmd*{#6}{%
        \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
      }%
    \ifcsdef{@GLS@user@#1@}%
    {%
      \PackageError{glossaries}%
      {Can't define '\string#7' as helper command
       '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
      {}%
    }%
    {%
      \expandafter\newcommand\expandafter*\expandafter
      {\csname @GLS@user@#1@\endcsname}[2][\{%

```

```

\new@ifnextchar[%
  {\csuse{@GLS@user@#1@}{##1}{##2}}%
  {\csuse{@GLS@user@#1@}{##1}{##2}[]}}%
\csdef{@GLS@user@#1@}##1##2[##3]{%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%
{%
  \PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand{\glsfieldxddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}

```

```

    }%
  }%
}
\newcommand{\glsfielddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \letcs#3{glo@\@glo@label @#2}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\ifglsfieldeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\ifglsfielddefeq}[5]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\ifdefstrequal
        \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
    }%
  }%
}

```

```

}%
{%
  \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\ifglsfieldcseq}[5]{%
  \glsdoifexists{#1}%
  {%
    \edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand*{\glswritedefhook}{%
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{}{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*{\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{}%
  {\longnewglossaryentry{#1}{#2}{#3}}%
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\let\glslabel\@glo@label
\let\@glo@name\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type
\let\@glo@symbol\@gls@symbol
\let\@glo@symbolplural\@gls@symbolplural
\let\@glo@text\@gls@text
\let\@glo@plural\@gls@plural
\let\@glo@first\@gls@first
\let\@glo@firstplural\@gls@firstplural
\let\@glo@sort\@gls@sort
\let\@glo@counter\@gls@counter
\def\@glo@see{ }%
\def\@glo@parent{ }%
\def\@glo@prefix{ }%
\@gls@initnonnumberlist
\def\@glo@useri{ }%
\def\@glo@userii{ }%
\def\@glo@useriii{ }%
\def\@glo@useriv{ }%
\def\@glo@userv{ }%
\def\@glo@uservi{ }%
\def\@glo@short{ }%
\def\@glo@shortpl{ }%
\def\@glo@long{ }%
\def\@glo@longpl{ }%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{%
  \ifignoredglossary\@glo@type
  {%
    \ifx\@glo@desc\@gls@desc

```

```

        \let\@glo@desc\@empty
    \fi
}%
{%
}%
\protected@edef\@glolist@\{ \csname glolist@\@glo@type\endcsname\}%
\expandafter\xdef\csname glolist@\@glo@type\endcsname{%
    \@glolist@\{ \@glo@label\},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
\else
    \ifdefequal\@glo@label\@glo@parent%
    {%
        \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
        \def\@glo@parent{%
        \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
        }%
        {%
            \ifglsentryexists{\@glo@parent}%
            {%
                \expandafter\xdef\csname glo@\@glo@label @parent\endcsname{%
                    \@glo@parent}%
                \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
                \advance\gls@level by 1\relax
                \ifx\@glo@name\@glsnoname
                    \expandafter\let\expandafter\@glo@name
                    \csname glo@\@glo@parent @name\endcsname
                    \ifx\@glo@plural\@gls@default@value
                        \expandafter\let\expandafter\@glo@plural
                        \csname glo@\@glo@parent @plural\endcsname
                    \fi
                \fi
            }%
        }%
        {%
            \PackageError{glossaries}%
            {%
                Invalid parent ‘\@glo@parent’
                for entry ‘\@glo@label’ - parent doesn’t exist%
            }%
            {%
                Parent entries must be defined before their children%
            }%
            \def\@glo@parent{%
            \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{%
            }%
        }%
    }%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%

```



```

\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs@glo@sort@glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx@glo@first@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef@glo@type@\@glo@type @counter}%
{%
  \def@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs@glo@defaultcounter@glo@type@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx@glo@name@glsnoname
  \glsnoname
  \let@glo@name@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef@glo@\@glo@label @numberlist}%
{%
  \csxdef@glo@\@glo@label @numberlist{%
    \noexpand@gls@missingnumberlist{\@glo@label}}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def@glo@@desc{\@glo@first}%
\ifx@glo@desc@glo@@desc

```

```

\let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
\@glsnodesc
\let\@glodesc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
\let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
\csname glo@\@glo@label @symbol\endcsname{
\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
\ifdefined\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\@do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%

```

```

\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}
\newcommand*{\glsmoveentry}[2]{%
  \edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
  \def\@glo@list{,%}
  \for@gl@entries[\@glo@type]{\@glo@label}%
  {%
    \if@defequal\@glo@thislabel\@glo@label
      {}{\@eappto\@glo@list{\@glo@label},}%
    }%
  \cslet\@glolist@\@glo@type{\@glo@list}%
  \csdef{glo@\@glo@thislabel @type}{#2}%
}
\if@gl@xindy
  \newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\if@gl@xindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\@subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \edef\@glo@esclabel{#1}%
  \@gl@checkmkidxchars\@glo@esclabel
  \protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
  \@gl@checkmkidxchars\@glo@sort
  \@gl@checkmkidxchars\@glo@prefix
  \edef\@glo@parent{\csname glo@#1@parent\endcsname}%
  \if@gl@xindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\@glo@parent\@empty

```

```

\@onelevel@sanitize\@glo@prefix
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
  \@glo@sort\@gls@actualchar\@glo@prefix
  \@glossaryentryfield{\@glo@esclabel}%
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
  \csname glo@\@glo@parent @index\endcsname\@gls@levelchar
  \@glo@sort\@gls@actualchar\@glo@prefix
  \@glossarysubentryfield
    {\csname glo@#1@level\endcsname}{\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
  \@ifpackageloaded{amsmath}%
  {\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
  {}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
  \ifmeasuring@
  \else
    #1%
  \fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
  \def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
  \ifdef\TX@trial
  {%
    \expandafter\@gls@patchtabularx\TX@trial{##1}!!%
    \let\glspatchtabularx\relax
  }%
  {}%
}
\newcommand*{\glsreset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsreset{#1}%
    }%
  }%
}
\newcommand*{\glslocalreset}[1]{%
  \gls@ifnotmeasuring
  {%

```

```

\glsdoifexists{#1}%
{%
    \@glslocalreset{#1}%
}%
}%
}
\newcommand*{\glsunset}[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glsunset{#1}%
        }%
    }%
}%
}
\newcommand*{\glslocalunset}[1]{%
    \gls@ifnotmeasuring
    {%
        \glsdoifexists{#1}%
        {%
            \@glslocalunset{#1}%
        }%
    }%
}%
}
\newcommand*{\@glslocalunset}{\@@glslocalunset}
\newcommand*{\@@glslocalunset}[1]{%
    \expandafter\let\csname ifglo@glstetoklabel{#1}@flag\endcsname\iftrue
}%
\newcommand*{\@glsunset}{\@@glsunset}
\newcommand*{\@@glsunset}[1]{%
    \expandafter\global\csname glo@glstetoklabel{#1}@flagtrue\endcsname
}%
\newcommand*{\@glslocalreset}{\@@glslocalreset}
\newcommand*{\@@glslocalreset}[1]{%
    \expandafter\let\csname ifglo@glstetoklabel{#1}@flag\endcsname\iffalse
}%
\newcommand*{\@glsreset}{\@@glsreset}
\newcommand*{\@@glsreset}[1]{%
    \expandafter\global\csname glo@glstetoklabel{#1}@flagfalse\endcsname
}%
\newcommand*{\glsresetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glsreset{\@glsentry}%
    }%
}%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glslocalreset{\@glsentry}%
    }%
}%

```

```

}%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*{\glsenableentrycount}{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*{newglossaryentry}[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*{\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
  \@glsreset{##1}%
  \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*{\@glslocalreset}[1]{%

```



```

\newcommand*{\@gls@write@entrycounts}{%
  \immediate\write\@auxout
  {\string\providecommand*\string\@gls@entry@count}[2]{}}%
\forallglsentries{\@glsentry}{%
  \ifglsused{\@glsentry}%
  {\immediate\write\@auxout
    {\string\@gls@entry@count{\@glsentry}{\glsentrycurrcount{\@glsentry}}}%
  }%
}%
}
\newcommand*{\@gls@entry@count}[2]{%
\newrobustcmd*{\cgl}{\@gls@hyp@opt\@cgl}
\newcommand*{\@cgl}[2][{}]{%
  \new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}}[{}]}%
}
\def\@cgl@#1#2[#3]{%
  \GlossariesWarning{\string\cgl\space is defaulting to
    \string\gls\space since you haven't enabled entry counting}%
  \@gls@{#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
  \ifglshaslong{#1}{\glsentrylong{#1}}{\glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cGls}{\@gls@hyp@opt\@cGls}
\newcommand*{\@cGls}[2][{}]{%
  \new@ifnextchar[{\@cGls@{#1}{#2}}{\@cGls@{#1}{#2}}[{}]}%
}
\def\@cGls@#1#2[#3]{%
  \GlossariesWarning{\string\cGls\space is defaulting to
    \string\Gls\space since you haven't enabled entry counting}%
  \@Gls@{#1}{#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylong{#1}}{\Glsentryfirst{#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
  \new@ifnextchar[{\@cglspl@{#1}{#2}}{\@cglspl@{#1}{#2}}[{}]}%
}
\def\@cglspl@#1#2[#3]{%
  \GlossariesWarning{\string\cglspl\space is defaulting to
    \string\glspl\space since you haven't enabled entry counting}%
  \@glspl@{#1}{#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
  \ifglshaslong{#1}{\glsentrylongpl{#1}}{\glsentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
  \new@ifnextchar[{\@cGlspl@{#1}{#2}}{\@cGlspl@{#1}{#2}}[{}]}%
}

```



```

}
\def\cGlspl@#1#2[#3]{%
  \GlossariesWarning{\string\cGlspl\space is defaulting to
    \string\Glspl\space since you haven't enabled entry counting}%
  \cGlspl@{#1}{#2}[#3]%
}
\newcommand*\cGlsplformat[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*\loadglsentries[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*\glstextformat[1]{#1}
\newcommand*\glsentryfmt{%
  \@@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*\@@gls@default@entryfmt[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \ifbool{glscompatible-3.07}%
        {%
          \protected@edef\@glo@etext{%
            #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
          }
          \xmakefirstuc\@glo@etext
        }%
        {%

```

```

        #2{\Glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
}%
{%
    \ifbool{glscompatible-3.07}%
    {%
        \protected@edef\@glo@etext{%
            #1{\glsentryfirstplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
        \xmakefirstuc\@glo@etext
    }%
    {%
        #1{\Glsentryfirstplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
    }%
}%
}%
{%
    \ifglsused\glslabel
    {%
        \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    }%
    {%
        \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
        {\glsentrydescplural{\glslabel}}%
        {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    }%
}%
}%
{%
    \glscapscase
    {%
        \ifglsused\glslabel
        {%
            #2{\glsentrytext{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}%
        }%
        {%
            #1{\glsentryfirst{\glslabel}}%
            {\glsentrydesc{\glslabel}}%
            {\glsentrysymbol{\glslabel}}{\glsinsert}%
        }%
    }%
}%

```

```

{%
  \ifglssused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #2{\Glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
  {%
    \ifbool{glscompatible-3.07}%
    {%
      \protected@edef\@glo@etext{%
        #1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
      \xmakefirstuc\@glo@etext
    }%
    {%
      #1{\Glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
  {%
    \ifglssused\glslabel
    {%
      \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
    {%
      \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
        {\glsentrydesc{\glslabel}}%
        {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    }%
  }%
  {%
    \ifglssused{\glslabel}%

```

```

    {%
      #2{\glscustomtext}%
      {\glentrydesc{\glslabel}}%
      {\glentrysymbol{\glslabel}}{%}%
    }%
    {%
      #1{\glscustomtext}%
      {\glentrydesc{\glslabel}}%
      {\glentrysymbol{\glslabel}}{%}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glusifplural
    {%
      \glscapscase
      {%
        \ifglused\glslabel
        {%
          \glentryplural{\glslabel}\glsinsert
        }%
        {%
          \glentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase

```

```

{%
  \ifglused\glslabel
  {%
    \glentrytext{\glslabel}\glsinsert
  }%
  {%
    \glentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \mfirstucMakeUppercase{\glentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase{\glentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\acronymfont{\glentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```

```

    }%
}%
{%
  \glscapscase
  {%
    \acronymfont{\glsentryshort{\glslabel}}\glsinsert
  }%
  {%
    \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
  }%
  {%
    \mfirstucMakeUppercase
    {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
  }%
}%
}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}%
{%
  \glscapscase
  {%
    \genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \mfirstucMakeUppercase
    {\genacrfullformat{\glslabel}{\glsinsert}}%
  }%
}%
}%
}%
{%
  \glscustomtext
}%
}

```

```

\newcommand*{\genacrfullformat}[2]{%
  \glentrylong{#1}#2\space
  (\protect\firstacronymfont{\glentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\genplacrfullformat}[2]{%
  \glentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glentryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\glsdisplayfirst}[4]{#1#4}
\newcommand*{\glsdisplay}[4]{#1#4}
\newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\glsdisplayfirst}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}
\newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}%
      {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\csuse{gls@#1@display}}%
      }%
    }%
  }%
  \@gls@doentrydef
}

```

```

    }%
    {%
        \noexpand\@gls@default@entryfmt
        {\noexpand\csuse{gls@#1@displayfirst}}%
        {\noexpand\glsdisplay}%
    }%
    }%
    }%
    \@gls@doentrydef
}
\define@key{glslink}{counter}{%
    \ifcsundef{c@#1}%
    {%
        \PackageError{glossaries}%
        {There is no counter called ‘#1’}%
        {%
            The counter key should have the name of a valid counter
            as its value%
        }%
    }%
    }%
    {%
        \def\@gls@counter{#1}%
    }%
}
\define@key{glslink}{format}{%
    \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
    \glslinkvar{#1}{#2}{#1}%
    \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
        you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
    \let\glslinkvar\@firstofthree
    \let\@gls@hyp@opt@cs#1\relax
    \@ifstar{\s@gls@hyp@opt}%
    {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@secondofthree
    \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
    \let\glslinkvar\@thirdofthree
    \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
    \@gls@hyp@opt\@gls@link
}

```



```

\newcommand*{\@gls@link}[3][\{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{\glsnumberformat}%
  \edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%

```

```

\glslinkpostsetkeys
\@gls@saveentrycounter
\@gls@setsort{\glslabel}%
\@do@wrglossary{#2}%
\ifKV@glslink@hyper
  \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
  \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
  \def\@gls@Hcounter{}%
  \ifthenelse{\equal{\@gls@counter}{equation}}{%
    {
      \ifcsundef{xatlevel@}%
      {%
        \edef\theglsentrycounter{\expandafter\noexpand
          \csname the\@gls@counter\endcsname}%
      }%
      {%
        \ifx\xatlevel@\@empty
          \edef\theglsentrycounter{\expandafter\noexpand
            \csname the\@gls@counter\endcsname}%
        \else
          \savecounters@
          \advance\c@equation by 1\relax
          \edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
          \ifcsundef{theH\@gls@counter}%
          {%
            \def\@gls@Hcounter{\theglsentrycounter}%
          }%
          {%
            \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
          }%
          \protected@edef\theHglentrycounter{\@gls@Hcounter}%
          \restorecounters@
        \fi
      }%
    }%
    {%
      \edef\theglsentrycounter{\expandafter\noexpand
        \csname the\@gls@counter\endcsname}%
    }%
    \ifx\@gls@Hcounter\@empty
      \ifcsundef{theH\@gls@counter}%
      {%
        \def\theHglentrycounter{\theglsentrycounter}%
      }%
    }%
  }%
}

```

```

}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1\relax
\def\@glo@prefix{()%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\@null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%

```

```

\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostsubst{%
  \noexpand\DTLsubstituteall\noexpand\gls@xdystring
  {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dostsubst
}%
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{%
      \expandafter\@gls@checkquote#1\@nil""\null
      \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
      \def\@gls@checkedmkidx{%
        \expandafter\@gls@checkescquote#1\@nil""\null
        \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
        \def\@gls@checkedmkidx{%
          \expandafter\@gls@checkescactual#1\@nil"??\null
          \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
          \def\@gls@checkedmkidx{%
            \expandafter\@gls@checkactual#1\@nil??\null
            \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
            \def\@gls@checkedmkidx{%
              \expandafter\@gls@checkbar#1\@nil||\null
              \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
              \def\@gls@checkedmkidx{%
                \expandafter\@gls@checkescbar#1\@nil|||\null
                \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                \def\@gls@checkedmkidx{%
                  \expandafter\@gls@checklevel#1\@nil!!\null
                  \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
                }
              }
            }
          }
        }
      }
    }
  \fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\@nil#2\@nil
    \ifx\@nil#3\@nil
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkquote{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
      \def\@gls@checkquote{\@gls@checkquote#3\@nil}%
    \fi
  }

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\""\@gls@quotechar
\@gls@quotechar\string\""\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\""\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\""\@gls@actualchar
\@gls@quotechar\string\""\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@

```

```

\@gls@quotechar\string\" \@gls@actualchar}%
\ifx\@null#3\@null
\def\@gls@checkescactual{\@gls@checkescactual#2\? \? \@null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\@null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@encapchar
\@gls@quotechar\string\" \@gls@encapchar}%
\def\@gls@checkescbar{\@gls@checkescbar#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@encapchar}%
\ifx\@null#3\@null
\def\@gls@checkescbar{\@gls@checkescbar#2\|\|\@null}%
\else
\def\@gls@checkescbar{\@gls@checkescbar#2\|#3\@null}%
\fi
\fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\@null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\@null#2\@null
\ifx\@null#3\@null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesclevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@levelchar
\@gls@quotechar\string\" \@gls@levelchar}%
\def\@gls@checkesclevel{\@gls@checkesclevel#3\@null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\" \@gls@levelchar}%
\ifx\@null#3\@null

```

```

\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!\!\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi

```

```

\fi
\@@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
\def\@@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@@gls@checkactual
}
\def\@gls@xdycheckquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@xdycheckquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"\string"}%
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"}%
\ifx\null#3\null
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\@@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%

```



```

\newcommand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\@backslashchar\@backslashchar
\newcommand\@gls@tmpb=\@backslashchar\@backslashchar
\newcommand\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\toks@={##1}%
\newcommand\ifx\@backslashchar\@backslashchar
\newcommand\ifx\@backslashchar\@backslashchar
\newcommand\edef\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\the\@gls@tmpb\@backslashchar\@backslashchar
\newcommand\def\@gls@xdycheckbackslash{\relax}%
\newcommand\else
\newcommand\edef\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\def\@gls@xdycheckbackslash{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\fi
\newcommand\else
\newcommand\edef\@gls@checkedmkidx{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\ifx\@backslashchar\@backslashchar
\newcommand\def\@gls@xdycheckbackslash{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\else
\newcommand\def\@gls@xdycheckbackslash{%
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}
\newcommand\fi
\newcommand\fi
\newcommand\@gls@xdycheckbackslash
}%
}
\def\@gls@xdycheckbackslash

\newlength\gls@tmplen
\newcommand*\@gls@dohypertarget}[2]{%
\@gls@showtarget{#1}%
\settoheight\gls@tmplen{#2}%
\raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*\@gls@dohyperlink}[2]{%
\@gls@showtarget{#1}%
\hyperlink{#1}{#2}%
}
\newcommand*\@gls@donohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
\let\@glslink\@gls@donohyperlink

```

```

}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
}
\def\@Gls@#1#2[#3]{%

```

```

\glsdoifexists{#2}%
{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\gl@sifplural\@secondoftwo
  \let\gl@scapscase\@secondofthree
  \let\gl@customtext\@empty
  \def\gl@insert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link{#1}{#2}{\@glo@text}%
  \ifKV@gl@link@local
    \gl@localunset{#2}%
  \else
    \gl@unset{#2}%
  \fi
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLS}{\@gl@hyp@opt\@GLS}
\newcommand*{\@GLS}[2][]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@secondoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link{#1}{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else
      \gl@unset{#2}%
    \fi
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@spl}{\@gl@hyp@opt\@gl@spl}
\newcommand*{\@gl@spl}[2][]{%
  \new@ifnextchar[{\@gl@spl@{#1}{#2}}{\@gl@spl@{#1}{#2}[]}%
}
\def\@gl@spl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@firstofthree
    \let\gl@customtext\@empty

```

```

\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\ifKV@glslink@local
  \glslocalunset{#2}%
\else
  \glsunset{#2}%
\fi
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\newcommand*{\@Glspl}[2][]{%
  \new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSpl}{\@gls@hyp@opt\@GLSpl}
\newcommand*{\@GLSpl}[2][]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else

```

```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newrobustcmd*{\glsdisp}{\@gls@hyp@opt\@glsdisp}
\newcommand*{\@glsdisp}[3][\@glsdoifexists{#2}]{%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \def\glscustomtext{#3}%
    \def\glsinsert{}%
    \def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
    \@gls@link{#1}{#2}{\@glo@text}%
    \ifKV@glslink@local
        \glslocalunset{#2}%
    \else
        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link{#1}{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glsstext}{\@gls@hyp@opt\@glsstext}
\newcommand*{\@glsstext}[2][\@new@ifnextchar[\@glsstext@{#1}{#2}]{\@glsstext@{#1}{#2}[]}]
\def\@glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][\@new@ifnextchar[\@GLStext@{#1}{#2}]{\@GLStext@{#1}{#2}[]}]
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glsstext}{\@gls@hyp@opt\@Glsstext}
\newcommand*{\@Glsstext}[2][\@new@ifnextchar[\@Glsstext@{#1}{#2}]{\@Glsstext@{#1}{#2}[]}]
\def\@Glsstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}

```

```

\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][]{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}]
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][]{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}]
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][]{%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]}]
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2][]{%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2}[]}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2][]{%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2][]{%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2}[]}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][]{%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][]{%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2][]{%

```

```

\new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} []}}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}}%
}
\newrobustcmd*{\@glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] []{%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} []}}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\@GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@GLSentryname{#2}#3}%
}
\newrobustcmd*{\@GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] []{%
  \new@ifnextchar[{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\@glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] []{%
  \new@ifnextchar[{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@GLSentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] []{%
  \new@ifnextchar[{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\@glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] []{%
  \new@ifnextchar[{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\@GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] []{%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar[{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\glssymbol}{\@gls@hyp@opt\@glssymbol}
\newcommand*{\@glssymbol}[2] [] {%
  \new@ifnextchar[{\@glssymbol@{#1}{#2}}{\@glssymbol@{#1}{#2} []}}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\Glsymbol}{\@gls@hyp@opt\@Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
  \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\GLSsymbol}{\@gls@hyp@opt\@GLSsymbol}
\newcommand*{\@GLSsymbol}[2] [] {%
  \new@ifnextchar[{\@GLSsymbol@{#1}{#2}}{\@GLSsymbol@{#1}{#2} []}}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbol{#2}#3}}%
}
\newrobustcmd*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2] [] {%
  \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2} []}}
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glsymbolplural}{\@gls@hyp@opt\@Glsymbolplural}
\newcommand*{\@Glsymbolplural}[2] [] {%
  \new@ifnextchar[{\@Glsymbolplural@{#1}{#2}}{\@Glsymbolplural@{#1}{#2} []}}
\def\@Glsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2] [] {%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2} []}}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}

```



```

\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][]{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\@Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\Glsuserii}{\@gls@hyp@opt\@Glsuserii}
\newcommand*{\@Glsuserii}[2][]{%
  \new@ifnextchar[{\@Glsuserii@{#1}{#2}}{\@Glsuserii@{#1}{#2}[]}}
\def\@Glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@Glsentryuserii{#2}#3}%
}
\newrobustcmd*{\GLSuserii}{\@gls@hyp@opt\@GLSuserii}
\newcommand*{\@GLSuserii}[2][]{%
  \new@ifnextchar[{\@GLSuserii@{#1}{#2}}{\@GLSuserii@{#1}{#2}[]}}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\@Glsentryuserii{#2}#3}}%
}
\newrobustcmd*{\glsuseriii}{\@gls@hyp@opt\@glsuseriii}
\newcommand*{\@glsuseriii}[2][]{%
  \new@ifnextchar[{\@glsuseriii@{#1}{#2}}{\@glsuseriii@{#1}{#2}[]}}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\Glsuseriii}{\@gls@hyp@opt\@Glsuseriii}
\newcommand*{\@Glsuseriii}[2][]{%
  \new@ifnextchar[{\@Glsuseriii@{#1}{#2}}{\@Glsuseriii@{#1}{#2}[]}}
\def\@Glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\@Glsentryuseriii{#2}#3}%
}
\newrobustcmd*{\GLSuseriii}{\@gls@hyp@opt\@GLSuseriii}
\newcommand*{\@GLSuseriii}[2][]{%
  \new@ifnextchar[{\@GLSuseriii@{#1}{#2}}{\@GLSuseriii@{#1}{#2}[]}}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\@Glsentryuseriii{#2}#3}}%
}
\newrobustcmd*{\glsuseriv}{\@gls@hyp@opt\@glsuseriv}
\newcommand*{\@glsuseriv}[2][]{%

```

```

\new@ifnextchar [{\@glsuseriv@{#1}{#2}}{\@glsuseriv@{#1}{#2} [] }}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\Glsuseriv}{\@gls@hyp@opt\@Glsuseriv}
\newcommand*{\@Glsuseriv}[2] [] {%
  \new@ifnextchar [{\@Glsuseriv@{#1}{#2}}{\@Glsuseriv@{#1}{#2} [] }}
\def\@Glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseriv{#2}#3}%
}
\newrobustcmd*{\GLSuseriv}{\@gls@hyp@opt\@GLSuseriv}
\newcommand*{\@GLSuseriv}[2] [] {%
  \new@ifnextchar [{\@GLSuseriv@{#1}{#2}}{\@GLSuseriv@{#1}{#2} [] }}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuseriv{#2}#3}}%
}
\newrobustcmd*{\glsuserv}{\@gls@hyp@opt\@glsuserv}
\newcommand*{\@glsuserv}[2] [] {%
  \new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} [] }}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] [] {%
  \new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} [] }}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} [] }}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} [] }}
\def\@GLSuservi@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}{#3}}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\ns@acrshort}[2] [] {%
  \new@ifnextchar[{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}%
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\acronymfont{\glsentryshort{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2][{}]{%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2}[]}%
}
\def\@acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2][{}]{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][{}]{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}%
}

```

```

\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glstryshortpl{#2}}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@acrlong{\@gl@hyp@opt\@ns@acrlong}
\newcommand*\@ns@acrlong[2][{}]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glstrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@Acrlong{\@gl@hyp@opt\@ns@Acrlong}
\newcommand*\@ns@Acrlong[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gls@hyp@opt\ns@ACRlong}
\newcommand*{\ns@ACRlong}[2][]{%
  \new@ifnextchar[{\@ACRlong{#1}{#2}}{\@ACRlong{#1}{#2}[]}%
}
\def\@ACRlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylong{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\ns@Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%

```

```

\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\ns@ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
  }%
  ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
  entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
  label and the field name}%
}%
}%
{%

```

```

    ??%
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
      \expandafter\Glsentrylong\@gls@rest
    \else
      \expandafter\ifx\@gls@body\glsentryshort\relax
        \expandafter\Glsentryshort\@gls@rest
      \else
        \expandafter\ifx\@gls@body\acronymfont\relax
          {%
            \let\glsentryshort\Glsentryshort
            \@glo@text
          }%
        \else
          \expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
            {%
              \let\glsentryshort\Glsentryshort
              \@glo@text
            }%
          \else
            \xmakefirstuc{\@glo@text}%
          \fi
        \fi
      \fi
    \fi
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newrobustcmd*{\Glsentrydesc}[1]{%
  \@Gls@entry@field{#1}{desc}%
}
\newcommand*{\glsentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}

```



```

\newrobustcmd*{\Glsentrydescplural}[1]{%
  \@Gls@entry@field{#1}{descplural}%
}
\newcommand*{\glsentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*{\Glsentrytext}[1]{%
  \@Gls@entry@field{#1}{text}%
}
\newcommand*{\glsentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*{\Glsentryplural}[1]{%
  \@Gls@entry@field{#1}{plural}%
}
\newcommand*{\glsentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*{\Glsentrysymbol}[1]{%
  \@Gls@entry@field{#1}{symbol}%
}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*{\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newrobustcmd*{\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*{\glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@Gls@value}%
    \xcapitalisewords{\@Gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsentrytitlecase}[2]{%
    \texorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\@Gls@entry@field{#1}{#2}}%
  }
}

```

```

    }
}
{
  \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*{\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newrobustcmd*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*{\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*{\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*{\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*{\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*{\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*{\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*{\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*{\glsentryservi}[1]{%
  \@gls@entry@field{#1}{servi}%
}
\newrobustcmd*{\Glsentryservi}[1]{%
  \@Gls@entry@field{#1}{servi}%
}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*{\Glsentryshort}[1]{%

```

```

    \@Gls@entry@field{#1}{short}%
}
\newcommand*{\glsentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}
\newrobustcmd*{\Glsentryshortpl}[1]{%
    \@Gls@entry@field{#1}{shortpl}%
}
\newcommand*{\glsentrylong}[1]{\@Gls@entry@field{#1}{long}}
\newrobustcmd*{\Glsentrylong}[1]{%
    \@Gls@entry@field{#1}{long}%
}
\newcommand*{\glsentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
\newrobustcmd*{\Glsentrylongpl}[1]{%
    \@Gls@entry@field{#1}{longpl}%
}
\newcommand*{\glsentryfull}[1]{%
    \acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
    \acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
    \acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
    \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
        \@Gls@entry@field{#1}{numberlist}%
    }%
}
\@ifpackageloaded{hyperref} {%
    \newcommand*{\glsdisplaynumberlist}[1]{%
        \GlossariesWarning
        {%
            \string\glsdisplaynumberlist\space
            doesn't work with hyperref.^^JUsing
            \string\glsentrynumberlist\space instead%
        }%
        \glsentrynumberlist{#1}%
    }%
}%
{\%
    \newcommand*{\glsdisplaynumberlist}[1]{%
        \glsdoifexists{#1}%
        {%
            \bgroup
            \edef\@glo@label{\glsdetoklabel{#1}}%
            \let\@org@glsnumberformat\glsnumberformat

```

```

\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
  \csname glo@\glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\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\glstildechar{\string~}
  \ifglsxindy
    \newcommand*{\@glsfirstletter}{A}
  \fi
  \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
    \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
  \ifglsxindy
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}%
      \@onelevel@sanitize\@glsfirstletter
    }
  \else
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
      \@GlsSetXdyFirstLetterAfterDigits
    }
  \fi
  \ifglsxindy
    \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
  \fi
  \newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
  }
  \ifglsxindy
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
      \renewcommand*{\@xdynumbergrouporder}{#1}%
    }
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%

```

```

\renewcommand*{\@xdynumbergrouporder}{#1}%
\@onelevel@sanitize\@xdynumbergrouporder
}
\else
\newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
\glsnoxywarning\GlsSetXdyNumberGroupOrder}
\newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
\@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglsxindy
\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
\renewcommand*{\@glsminrange}{#1}}
\else
\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\@glscataddpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@glsc@classI:=\@glsc@xdy@locationlist\do{%
\protected@write\glswrite{}{(define-location-class
\string"\@glsc@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @glsc@xdy@Lclass@\@glsc@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```

```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}\{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "{{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glsseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string

```

```

\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glresetentrylist
\glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
\string"\glsoopenbrace\string\glossaryentrynumbers
\glsoopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsoopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string"}}%

```



```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax
\ifundef{\glswrite}{\newwrite\glswrite}{}\relax
\openout\glswrite=\istfilename
\write\glswrite{\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"glsnumbers\string"}
\write\glswrite{page_compositor \string"glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
  \@ifpackageloaded{tracklang}%
  {%
    \IfTrackedLanguage{german}%
    {%
      \def\@gls@extramakeindexopts{-g}%
    }%
  }%
}%
\edef\@gls@docheckquotedef{%
  \noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
    \noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
    \noexpand\toks@={####1}%
  }%
}

```

```

\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{%
\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesobar###1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel###1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%

```

```

\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote####2\expandonce{\csname#1\endcsname}%
####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkescquote
}%
}%
\@gls@docheckquotedef
}
\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
{\string\GlsSetQuote\space not permitted here}%
{Move \string\GlsSetQuote\space earlier in the preamble, as
soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*{\@gls@extramakeindexopts}[1]{%
\newcommand{\noist}{%
\@gls@addpredefinedattributes
\let\writeist\relax
}
\newcommand*{\@makeglossary}[1]{%
\ifglossaryexists{#1}%
{%
\ifglssavewrites
\expandafter\newtoks\csname glo@#1@filetok\endcsname
\else
\expandafter\newwrite\csname glo@#1@file\endcsname
\expandafter\@gls@openfile\csname glo@#1@file\endcsname{#1}%
\fi
\@gls@renewglossary
\writeist
}%
{%
\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@gls@openfile}[2]{%
\immediate\openout#1=\jobname.\csname @glotype@#2@out\endcsname
\PackageInfo{glossaries}{Writing glossary file
\jobname.\csname @glotype@#2@out\endcsname}%
}
\newcommand*{\@closegls}[1]{%
\closeout\csname glo@#1@file\endcsname
}
\ifglxsindy
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}

```

```

{%
  \@closegls{#1}%
  \ifdefstring{\glsorder}{letter}%
    {\def\@gls@order{-M ord/letorder }}%
    {\let\@gls@order\empty}%
  \ifcsundef{@xdy@#1@language}%
    {\let\@gls@langmod\@xdy@main@language}%
    {\letcs\@gls@langmod{@xdy@#1@language}}%
  \edef\@gls@dothiswrite{\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \@gls@istfilebase\space
    -C \@gls@codepage\space
    -t \jobname.\csuse{@glotype@#1@log}
    -o \jobname.\csuse{@glotype@#1@in}
    \jobname.\csuse{@glotype@#1@out}}}%
 }%
  \@gls@dothiswrite
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
  \ifglossaryexists{#1}
  {%
    \@closegls{#1}%
    \ifdefstring{\glsorder}{letter}%
      {\def\@gls@order{-l }}%
      {\let\@gls@order\empty}%
    \edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
      -s \istfilename\space
      -t \jobname.\csuse{@glotype@#1@log}
      -o \jobname.\csuse{@glotype@#1@in}
      \jobname.\csuse{@glotype@#1@out}}}%
   }%
    \@gls@dothiswrite
 }%
  {\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\fi
\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}%

```

```

{%
  \ifdefstring{\glsorder}{letter}%
  {\def\@gls@order{-M ord/letorder }}%
  {\let\@gls@order\@empty}%
  \ifcsundef{xdy@#1@language}%
  {\let\@gls@langmod\@xdy@main@language}%
  {\letcs\@gls@langmod{xdy@#1@language}}}%
  \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
    -I xindy
    \@gls@order
    -L \@gls@langmod\space
    -M \gls@istfilebase\space
    -C \gls@codepage\space
    -t \jobname.\csuse{@glotype@#1@log}
    -o \jobname.\csuse{@glotype@#1@in}
    \jobname.\csuse{@glotype@#1@out}}}%
  }%
  \@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
  doesn't exist. Rerun may be required}}}%
}%
{\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
  \ifglossaryexists{#1}
  {%
    \IfFileExists{\jobname.\csuse{@glotype@#1@out}}{%
      {%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-l }}%
        {\let\@gls@order\@empty}%
        \edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
          -s \istfilename\space
          -t \jobname.\csuse{@glotype@#1@log}
          -o \jobname.\csuse{@glotype@#1@in}
          \jobname.\csuse{@glotype@#1@out}}}%
        }%
        \@gls@dothiswrite
      }%
      {\GlossariesWarning{can't automake '#1': \jobname.\csuse{@glotype@#1@out}
        doesn't exist. Rerun may be required}}}%
    }%
    {\GlossariesWarning{Can't make glossary '#1', it doesn't exist}}%
  }%
}
}

```

```

\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
\@gls@@automake@immediate
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{}%
\protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{}%
\ifundef\@gls@extramakeindexopts
{}%
}%
\protected@write\@auxout{}{\string\providecommand
\string\@gls@extramakeindexopts[1]{}%
\protected@write\@auxout{}{\string\@gls@extramakeindexopts
{\@gls@extramakeindexopts}}%
}%
\protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
\protected@write\@auxout{}{\string\@glsorder{\glsorder}}%
\@for\@glo@type:=\@glo@types\do{%
\ifthenelse{\equal{\@glo@type}{} }{}{%
\@makeglossary{\@glo@type}}%
}%
\renewcommand*\newglossary[4][]{%
\PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
\let\@makeglossary\@gobble
\let\makeglossaries\relax
\@disable@onlypremakeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\def\warn@noprintglossary{%
\ifdefstring{\@glo@types}{,}%
{%
\GlossariesWarningNoLine{No glossaries have been defined}%
}%
}%

```



```

\GlossariesWarningNoLine{No \string\printglossary\space
or \string\printglossaries\space
found. ^^J(Remove \string\makeglossaries\space if you
don't want any glossaries.) ^^JThis document will not
have a glossary}%
}%
}%
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser
{\noexpand\glsnumlistparser}{\delimN}}}%
\@gls@dodolistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
\let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
\renewcommand*{\@gls@doautomake}{%
\@for\@gls@type:=\@glo@types\do{%
\ifdefempty{\@gls@type}{}%
{\@gls@automake{\@gls@type}}}%
}%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\AtEndDocument{%
\warn@nomakeglossaries
\warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
\@domakeglossaries
{%
\renewcommand{\@gls@noref@warn}[1]{%
\GlossariesWarning{Empty glossary for
\string\printnoidxglossary[type={##1}].
Rerun may be required (or you may have forgotten to use
commands like \string\gls)}%
}%
\let\@gls@checkmkidxchars\@gobble
\glsesclocationsfalse
\let\@do@wrglossary\gls@noidxglossary
\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
\edef\@gls@label{\glsdetoklabel{##1}}}%
\protected@write\@auxout{}{%
\string\@gls@reference

```

```

        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}%
        {%
            \string\glsseeformat##2}%
        }%
    }%
}%
\AtBeginDocument
{%
    \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
    \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@assign@sortkey
    \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*{\new@glossaryentry[2]{%
    \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
    {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
    \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
    \ifdef\@gls@loclist
    {%
        \glsnoidxloclist{\@gls@loclist}%
    }%
    {%
        ??\glsdoifexists{##1}%
        {%
            \GlossariesWarning{Missing location list for '##1'. Either
            a rerun is required or you haven't referenced the entry}%
        }%
    }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
    \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
    \ifdef\@gls@loclist
    {%
        \def\@gls@noidxloclist@sep{%
            \def\@gls@noidxloclist@sep{%

```

```

\def\@gls@noidxloclist@sep{%
  \glsnumlistsep
}%
\def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
}%
}%
\def\@gls@noidxloclist@finalsep{}%
\def\@gls@noidxloclist@prev{}%
\forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
\@gls@noidxloclist@finalsep
\@gls@noidxloclist@prev
}%
{%
??\glsdoifexists{##1}%
{%
  \GlossariesWarning{Missing location list for ‘##1’. Either
    a rerun is required or you haven’t referenced the entry}%
}%
}%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org\glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org\glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org\glsnoidxdisplayloc
  \let\glsseeformat\@gls@org\glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%

```

```

}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
  \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
  together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
  is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
    {\@glo@counterprefix}%
    {\@gls@counter}%
    {\@glsnumberformat}%
    {\@glslocref}%
    }%
  }%
}
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \fi
  }%
}

```

```

        \fi
    \else
        \@glsopenfile{\glswrite}{\@glo@type}%
        \immediate\write\glswrite{%
            \expandafter\the
            \csname glo@\@glo@type @filetok\endcsname}%
        \immediate\closeout\glswrite
    \fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp~J}%
    \else
        \ifcsdef{glo@#1@file}%
        {%
            \expandafter\protected@write\csname glo@#1@file\endcsname{%
                \gls@disablepagerefexpansion}{#2}%
            }%
        {%
            \ifignoredglossary{#1}{}%
            {%
                \GlossariesWarning{No file defined for glossary ‘#1’}%
            }%
        }%
    \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*{\glswriteentry}[2]{%
    \ifglsexonlyfirst
        \ifglssused{#1}{#2}%
    \else
        #2%
    \fi
}

```

```

\fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\for\@gls@this:=\gls@protected@pagefmts\do
{%
\expandafter\let\@gls@this\relax
}%
}
\newcommand*{\gls@alphpage}{\@alph\c@page}
\newcommand*{\gls@Alphpage}{\@Alph\c@page}
\newcommand*{\gls@numberpage}{\number\c@page}
\newcommand*{\gls@arabicpage}{\@arabic\c@page}
\newcommand*{\gls@romanpage}{\romannumeral\c@page}
\newcommand*{\gls@Romanpage}{\@Roman\c@page}
\newcommand*{\glsaddprotectedpagefmt}[1]{%
\ea\pt\o\gls@protected@pagefmts{\expandonce{\csname gls#1page\endcsname}}%
\csedef{gls#1page}{\expandonce{\csname#1\endcsname}\noexpand\c@page}%
\ea\pt\o\@wrglossarynumberhook{%
\noexpand\let\expandonce{\csname org@gls#1\endcsname}%
\expandonce{\csname#1\endcsname}%
\noexpand\def\expandonce{\csname#1\endcsname}{%
\noexpand\@wrglossary@pageformat
\expandonce{\csname gls#1page\endcsname}%
\expandonce{\csname org@gls#1\endcsname}%
}%
}%
}
\newcommand*\@wrglossarynumberhook{}
\newcommand{\@wrglossary@pageformat}[3]{%
\ifx#3\c@page #1\else #2#3\fi
}
\newcommand*{\@do@wrglossary}[1]{%
\ifglslsclocations
\@do@esc@wrglossary{#1}%
\else
\@do@noesc@wrglossary{#1}%
\fi
}
\newcommand*{\@do@noesc@wrglossary}[1]{%
\expandafter\def\expandafter\@glslocref\expandafter{\theglentrycounter}%
\expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
\ifx\@glsHlocref\@glslocref
\def\@glo@counterprefix{%
\else
\protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@glslocref}{\@glsHlocref}%
}%
\@do@gls@getcounterprefix

```

```

\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@gls@locref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@gls@locref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
\def\@glo@counterprefix{%
\else
\protected@edef\@gls@Hlocref{\theHglentrycounter}%
\@gls@checkmkidxchars\@gls@Hlocref
\edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
{\@gls@locref}{\@gls@Hlocref}%
}%
\@do@gls@getcounterprefix
\fi
\edef\@gls@label{\glsdetoklabel{#1}}%
\@@do@wrglossary

```

```

}
\newcommand*{\@@do@@wrglossary}{%
  \ifglxsindy
    \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
    \def\@glo@range{}%
    \expandafter\if\@glo@prefix(\relax
      \def\@glo@range{:open-range}%
    \else
      \expandafter\if\@glo@prefix)\relax
      \def\@glo@range{:close-range}%
    \fi
  \fi
  \gls@glossary{csname glo@\@gls@label @type\endcsname}{%
    (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
      :locref \string"\@glo@counterprefix{\@glslocref}\string" %
      :attr \string"\@gls@counter\@glo@suffix\string"
      \@glo@range
    )
  }%
\else
  \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
  {\@glo@counterprefix}%
  \gls@glossary{csname glo@\@gls@label @type\endcsname}{%
    \string@glossaryentry{csname glo@\@gls@label @index\endcsname
      \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
  \fi
}
\newcommand*\@gls@getcounterprefix[2]{%
  \edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
    \ifx\@glo@counterprefix\@empty
      \GlossariesWarning{Hyper target ‘#2’ can’t be formed by
        prefixing^^Jlocation ‘#1’. You need to modify the
        definition of \string\theH\@gls@counter^^Jotherwise you
        will get the warning: “‘name{\@gls@counter.#1}’ has been^^J
        referenced but does not exist”}%
    \fi
  \fi
}

```



```

\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :key (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*{\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*{\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*{\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*{\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }

```

```

\newrobustcmd*{\glsseeitem}[1]{\gls hyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{\glsentrytext{#1}}
\newcommand*{\gls@save@numberlist}[1]{%
  \ifglssavenumberlist
    \toks@{#1}%
    \edef\@do@writeaux@info{%
      \noexpand\csgdef{glo@\glscurrententrylabel @numberlist}{\the\toks@}%
    }%
    \@onelevel@sanitize\@do@writeaux@info
    \protected@write\@auxout{}\@do@writeaux@info%
  \fi
}
\newcommand*{\warn@noprntglossary}{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*{\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*{\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*{\printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*{\printnoidxglossaries}{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*{\@printgloss@setsort}{%
}
\newcommand*{\@gls@preglossaryhook}{%
}
\newcommand{\@printglossary}[2]{%

```

```

\def\@glo@type{\glsdefaulttype}%
\def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
\def\glossarytoctitle{\glossarytitle}%
\let\org@glossarytitle\glossarytitle
\def\@glossarystyle{%
  \ifx\@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '\@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def\gls@dotoc@title{\glssettoctitle{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\bgroup
\@printgloss@setsort
\setkeys{printgloss}{#1}%
\@printgloss@checkexists{\@glo@type}%
{%
  \ifx\glossarytitle\org@glossarytitle
  \else
    \expandafter\let\csname @glo@type\@glo@type @title\endcsname
      \glossarytitle
  \fi
  \let\currentglossary\@glo@type
  \let\org@glossaryentrynumbers\glossaryentrynumbers
  \let\glsnonextpages\@glsnonextpages
  \let\glsnextpages\@glsnextpages
  \let\nopostdesc\@nopostdesc
  \gls@dotoc@title
  \@glossarystyle
  \let\gls@org@glossaryentryfield\glossentry
  \let\gls@org@glossarysubentryfield\subglossentry
  \renewcommand{\glossentry}[1]{%
    \xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
    \gls@org@glossaryentryfield{##1}%
  }%
  \renewcommand{\subglossentry}[2]{%
    \xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
    \gls@org@glossarysubentryfield{##1}{##2}%
  }%
  \@gls@preglossaryhook
  #2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}

```

```

\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
  {}%
  {\null}%
  \ifglxindy
    \ifcsundef{@xdy@\@glo@type @language}%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\@xdy@main@language}%
          }%
        }%
      }%
    }%
    {%
      \edef\@do@auxoutstuff{%
        \noexpand\AtEndDocument{%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\providecommand\string\@xdylanguage[2]{}%
          }%
          \noexpand\immediate\noexpand\write\@auxout{%
            \string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
              @language\endcsname}%
          }%
        }%
      }%
    }%
  }%
  \@do@auxoutstuff
  \edef\@do@auxoutstuff{%
    \noexpand\AtEndDocument{%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\providecommand\string\@gls@codepage[2]{}%
      }%
      \noexpand\immediate\noexpand\write\@auxout{%
        \string\@gls@codepage{\@glo@type}{\@gls@codepage}%
      }%
    }%
  }%
  \@do@auxoutstuff
}
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
  \GlossariesWarningNoLine{\string\makeglossaries\space
    hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
  \glosortentrieswarning
  \def\@glo@sortinglist{}%
  \def\@glo@sortinghandler{#1}%
  \edef\@glo@type{#2}%

```

```

\forlistcsloop{\@glo@do@sortentries}{@glsref@#2}%
\csdef{@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
  \xifinlistcs{\@this@label}{@glsref@#2}%
  {%
    \listcsxadd{@glsref@#2}{\@this@label}%
  }%
  \ifcsdef{@glo@sortingchildren@\@this@label}%
  {%
    \@glo@addchildren{#2}{\@this@label}%
  }%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
  \bgroup
  \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
  \@for\@this@childlabel:=\@glo@childlist\do
  {%
    \xifinlistcs{\@this@childlabel}{@glsref@#1}%
    {%
      \listcsxadd{@glsref@#1}{\@this@childlabel}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
    {%
      \@glo@addchildren{#1}{\@this@childlabel}%
    }%
    {%
      }%
    }%
  }%
  \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
  \ifglshasparent{#1}%
  {%
    \edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
    \ifcsundef{@glo@sortingchildren@\@glo@parent}%
    {%
      \csdef{@glo@sortingchildren@\@glo@parent}{}%
    }%
    {%
      }%
    }%
    \expandafter\@glo@sortedinsert
    \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
    \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
    {%
      }%
    {%
      \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
  }%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@glsetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@glsetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@word}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=word]}{}%
}%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\cuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%

```

```

\ifdefstring{\@glo@default@sorttype}{standard}%
{%
  \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forgl@sentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
      }%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@{\@glo@type}%
  }%
  {%
    \listcsadd{\@glsref@{\@glo@type}}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  }%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@{\@glo@type}%
  }%
  \ifcsdef{\@glo@sortmacro@{\@glo@sorttype}%
  }%
  \csuse{\@glo@sortmacro@{\@glo@sorttype}}{\@glo@type}%
  }%
  {%
    \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
  }%
  \glossarysection[\glossarytoctitle]{\glossarytitle}%
  \glossarypreamble
  \def\@gls@currentlettergroup{}%
  \begin{theglossary}%

```



```

\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gls@noidx@do}{\@glsref@{\@glo@type}%
\end{theglossary}}%
\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\uccode'#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@='#1\relax
}%
{%
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{glo@\glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
}

```

```

    {%
      \subglossentry{\gls@level}{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
  {%
    \letcs{\@gls@sort}{glo\glsdetoklabel{#1}@sort}%
    \expandafter\glo@grabfirst\@gls@sort{}\@nil
    \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
    {%
      \ifdefempty{\@gls@currentlettergroup}{}%
      {%
        \global\let\@glo@thislettergrp\@glo@thislettergrp
        \glsgroupskip
      }%
      \glsgroupheading{\@glo@thislettergrp}%
    }%
    \global\let\@gls@currentlettergroup\@glo@thislettergrp
    \ifdefvoid{\@gls@loclist}
    {%
      \glossentry{#1}{}%
    }%
    {%
      \glossentry{#1}%
      {%
        \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
      }%
    }%
  }%
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%

```

```

\ifdefstring{\@gls@noidxloclist@prev}{#1}%
{%
}%
{%
\@gls@noidxloclist@sep
\@gls@noidxloclist@prev
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
\setentrycounter{#1}{#2}%
\csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
\glsdoifexistsorwarn{#2}%
{%
\ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
\ifinlistcs{#2}{\@glsref@#1}%
{}%
{\listcsgadd{\@glsref@#1}{#2}}%
\ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
{\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
{}%
\listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
}%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotoc@title\relax
}
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotoc@title\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{\@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossentrycompatibility
\csname @glsstyle@#1\endcsname}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%

```

```

\ifcase\gls@numberedsection@nr\relax
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{}%
\or
  \renewcommand*{\@@glossarysecstar}{}%
  \renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
  \renewcommand*{\@@glossarysecstar}{*}%
  \renewcommand*{\@@glossaryseclabel}{%
    \protected@edef\@currentlabelname{\glossarytoctitle}%
    \label{\glsautoprefix\@glo@type}}%
  \fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glssubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
    \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}

```

```

}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglsentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglsentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%

```

```

\ifglssubentrycounter
  \theglossarysubentry)\space
\fi
}
\newcommand*{\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetsubentrycounter
  \fi
}
\newcommand*{\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\providecommand*{\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@glossentry
}
\newcommand*{\glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glssdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glssdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}

```

```

\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\providecommand*{\compatiblesubglossentry}[3]{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
    {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
    {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
    {\the\toks@}}%
  }%
  \@do@subglossentry
}
\newcommand*{\setglossentrycompatibility}{%
  \let\glossentry\compatibleglossentry
  \let\subglossentry\compatiblesubglossentry
}
\setglossentrycompatibility
\newcommand{\glossaryentryfield}[5]{%
  \GlossariesWarning
  {Deprecated use of \string\glossaryentryfield.^^J
    I recommend you change to \string\glossentry.^^J
    If you've just upgraded, try removing your gls auxiliary
    files^^J and recompile}%
  \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
\newcommand*{\glossarysubentryfield}[6]{%
  \GlossariesWarning

```

```

{Deprecated use of \string\glossarysubentryfield.^^J
 I recommend you change to \string\subglossentry.^^J
 If you've just upgraded, try removing your gls auxiliary
 files^^J and recompile}%
\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
 \@gls@getgrouptitle{#1}{\@gls@grptitle}%
 \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
 \dtl@ifsingle{#1}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
{%
 \ifboolexpr{test{\ifstrequal{#1}{\glssymbols}}
 or test{\ifstrequal{#1}{\glsnumbers}}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
{%
 \def#2{#1}%
}%
}%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
 \DTLifint{#1}%
 {\edef#2{\char#1\relax}}%
 {%
 \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
}%
}
\newcommand*{\glsgetgrouplabel}[1]{%
 \ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
 \ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
}
\newcommand*{\setentrycounter}[2][1]{%
 \def\@glo@counterprefix{#1}%
 \ifx\@glo@counterprefix\@empty
 \def\@glo@counterprefix{.}%
 \else
 \def\@glo@counterprefix{.#1}%
 \fi
 \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
 \ifcsundef{@glsstyle@#1}%
 {%
 \PackageError{glossaries}{Glossary style '#1' undefined}{}%

```



```

}%
{%
  \csname @glsstyle@#1\endcsname
}%
\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^~J
      I recommend you switch to \string\setglossarystyle\space unless
      you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\csuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%

```

```

}%
{%
  \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
    \else
      \@delimR#1\delimR\delimR\%
    \fi
    \ifx\#2\%
    \else
      #2%
    \fi
    \ifx\#3\%
    \else
      \@glshypernumber#3\@nil
    \fi
  }
  \def\@delimR#1\delimR #2\delimR #3\{%
  \ifx\#2\%
    \@delimN{#1}%
  \else
    \@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
  \fi}
  \def\@delimN#1{\@delimN#1\delimN \delimN\}
  \def\@delimN#1\delimN #2\delimN#3\{%
  \ifx\#3\%
    \@gls@numberlink{#1}%
  \else
    \@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
  \fi
  }
  \def\@gls@numberlink#1{%
  \begingroup
  \toks@={}%
  \@gls@removespaces#1 \@nil
  \endgroup}
  \def\@gls@removespaces#1 #2\@nil{%
  \toks@=\expandafter{\the\toks@#1}%
  \ifx\#2\%
    \edef\x{\the\toks@}%
    \ifx\x\empty
    \else
      \hyperlink{\glstrycounter\@gls@counterprefix\the\toks@}%
        {\the\toks@}%
    \fi
  \else
    \@gls@ReturnAfterFi{%
      \@gls@removespaces#2\@nil
    }%
  }%

```

```

\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
    }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*\acrpluralsuffix{\glsacrpluralsuffix}
\newrobustcmd*\glstextup[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*\glsshortkey{short}
\newcommand*\glsshortpluralkey{shortplural}
\newcommand*\glslongkey{long}
\newcommand*\glslongpluralkey{longplural}
\newrobustcmd*\acrfull{\@gls@hyp@opt\@ns@acrfull}
\newcommand*\@ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}{%
    {\@acrfull{#1}{#2}}}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*\acrfullfmt[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}{#5}{#2}{#3}{#4}}%
}

```

```

\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}{%
    {\@Acrfull{#1}{#2}[]}%
  }
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}{%
    {\@ACRfull{#1}{#2}[]}%
  }
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
    {\@acrfullpl{#1}{#2}[]}%
  }
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
    {\@Acrfullpl{#1}{#2}[]}%
  }
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
    {\@ACRfullpl{#1}{#2}[]}%
  }
}

```

```

}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand*\@ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\@acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*\@newacronymhook{}
\newcommand*\@SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acrentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
  }%
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \gslongtok{##4}%
  \newacronymhook
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*\@acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}%
\renewcommand*\@Acrfullfmt}[3]{%

```

```

\glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
\ifcsundef{@glsacr@dispstyle@#1}
{%
\PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
}%
{%
\ifdefempty{\@glsacronymlists}%
{%
\DeclareAcronymList{\acronymtype}%
}%
{}%
\SetGenericNewAcronym
\GlsUseAcrStyleDefs{#1}%
\@for\@gls@type:=\@glsacronymlists\do{%
\defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}}%
}%
}%
}
\newcommand*{\newacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
}%
{%
\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
}
\newcommand*{\renewacronymstyle}[3]{%
\ifcsdef{@glsacr@dispstyle@#1}%
{%

```

```

\csdef{@glsacr@dispstyle@#1}{#2}%
\csdef{@glsacr@styledefs@#1}{#3}%
}%
{%
\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsacrentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glsentryshort{##1}})%
}

```

```

}%
\renewcommand*{\genplacrfullformat}[2]{%
\glentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})%}
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}

```



```

}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glssacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sp-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\glspacespace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%

```

```

{%
  \GlsUseAcrEntryDisplayStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {%
    {\glscustomtext\glsinsert}%
  }%
}

```

```

{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\glsentrylong{##2}##3\space
(\acronymfont{\glsentryshort{##2}})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\mfirstucMakeUppercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
\glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
\Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentrylong{##1}}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%

```

```

{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \glshyperfirstfalse
  \renewcommand*{\genacrfullformat}[2]{%
    \protect\firstacronymfont{\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}{\glscrpluralsuffix}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
      (\glsentrylong{##2})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
      (\glsentrylong{##2})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
        (\glsentrylong{##2})}}%
  }%
  \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}%
  }%
  \renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
      (\glsentrylongpl{##2})}%
  }%
  \renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
        (\glsentrylongpl{##2})}}%
  }%
  \renewcommand*{\glsentryfull}[1]{%
    \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
  \renewcommand*{\Glsentryfull}[1]{%
    \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%

```

```

\renewcommand*{\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\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]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\acrlong
  \let\aclp\acrlongpl
  \let\Acl\Acrlong
  \let\Aclp\Acrlongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglssacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    }
  }

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDefaultAcronymStyle}{%
    \@for\@gls@type:=\@gls@acronymlists\do{%
        \SetDefaultAcronymDisplayStyle{\@gls@type}%
    }%
    \renewcommand{\newacronym}[4][]{%
        \ifx\@gls@acronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetDefaultAcronymDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \DefaultNewAcronymDef
    }%
    \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
    \footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrno linkfootnote}[3]{%
    \footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
    \def\glsentryfmt[1]{%
        \ifdefempty\glscustomtext

```



```

{%
  \ifglused{\glslabel}%
  {%
    \acronymfont{\gls Genentryfmt}%
  }%
  {%
    \firstacronymfont{\gls Genentryfmt}%
    \ifglshassymbol{\glslabel}%
    {%
      \expandafter\protect\expandafter\acrfootnote\expandafter
      {\@gls@link@opts}{\@gls@link@label}%
      {%
        \glsifplural
        {\glsentrysymbolplural{\glslabel}}%
        {\glsentrysymbol{\glslabel}}%
      }%
    }%
  }%
}
}%
{\gls customtext\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\gls longtok},%
      longplural={\the\gls longtok\noexpand\acrpluralsuffix},%
      symbol={\the\gls longtok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@symbolplural\gls@assign@symbolplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%

```

```

}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
      \def\@gls@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@gls@type}%
      \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else
    \ifglsacrsmaller
      \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
    \fi
  \fi
  \ifglsacrdua
    \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
      can’t both be set}{}%
  \fi
}%
\newcommand*{\SetDescriptionDUAacronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\DescriptionDUANewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},

```

```

text={\the\glslongtok},%
first={\the\glslongtok},%
plural={\noexpand\expandonce\noexpand\@glo@longpl},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \ifglsacrsmaller
      \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
        can’t both be set}{}%
    \fi
  \fi
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%

```

```

\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsentryfmt}\gls@org@insert
}%
{%
\glsentryfmt
\ifglshassymbol{\glslabel}%
{%
\glsifplural
{%
\def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
}%
{%
\def\@glo@symbol{\glsentrysymbol{\glslabel}}%
}%
\space(\protect\firstacronymfont
{\glscapscase
{\@glo@symbol}
{\@glo@symbol}
{\mfirstucMakeUppercase{\@glo@symbol}}})%
}%
}%%
}%
\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
sort={\the\glsshorttok},%
first={\the\glslongtok},%

```

```

firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
  \renewcommand{\newacronym}[4][\]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDescriptionAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionNewAcronymDef
  }%
  \for\@gls@type:=\@glsacronymlists\do{%
    \SetDescriptionAcronymDisplayStyle{\@gls@type}%
  }%
  \ifglsacrsmallcaps
    \renewcommand{\acronymfont}[1]{\textsc{##1}}
    \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
  \else

```

```

\ifglsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\ifglused{\glslabel}%
{%
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
{%
\firstacronymfont{\glsgenentryfmt}\gls@org@insert
\ifglshaslong{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
}%
\glsifplural
{\glentrylongpl{\glslabel}}%
{\glentrylong{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%

```

```

        \the\glskeylisttok
    }%
}%
\let\@org@gl@s@assign@plural\gl@s@assign@plural
\let\@org@gl@s@assign@firstpl\gl@s@assign@firstpl
\let\@org@gl@s@assign@descplural\gl@s@assign@descplural
\def\gl@s@assign@firstpl##1##2{%
    \@@gl@s@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@s@assign@plural##1##2{%
    \@@gl@s@expand@field{##1}{plural}{##2}%
}%
\def\gl@s@assign@descplural##1##2{%
    \@@gl@s@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gl@s@assign@plural\@org@gl@s@assign@plural
\let\gl@s@assign@firstpl\@org@gl@s@assign@firstpl
\let\gl@s@assign@descplural\@org@gl@s@assign@descplural
}
\newcommand*{\SetFootnoteAcronymStyle}{%
    \renewcommand{\newacronym}[4][{}]{%
        \ifx\@gl@sacronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetFootnoteAcronymDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \FootnoteNewAcronymDef
    }%
    \@for\@gl@s@type:=\@gl@sacronymlists\do{%
        \SetFootnoteAcronymDisplayStyle{\@gl@s@type}%
    }%
    \ifgl@sacrsmallcaps
        \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
        \renewcommand*{\acrpluralsuffix}{\glssupacrpluralsuffix}%
    \else
        \ifgl@sacrsmaller
            \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
        \fi
    \fi
    \ifgl@sacrdua
        \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
            can’t both be set}{}%
    \fi

```

```

}%
\DeclareRobustCommand*{\glsdoparenifnotempty}[2]{%
  \protected@edef\gls@tmp{#1}%
  \ifdefempty\gls@tmp
  {}%
  {%
    \ifx\gls@tmp\@gls@default@value
    \else
    \space (#2{#1})%
    \fi
  }%
}
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\glscustomtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\glsentryfmt}\gls@org@insert
      }%
      {%
        \glsentryfmt
        \ifglshassymbol{\glslabel}%
        {%
          \glsifplural
          {%
            \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
          }%
          \space
          (\glscapscase
            {\firstacronymfont{\@glo@symbol}}%
            {\firstacronymfont{\@glo@symbol}}%
            {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
          }%
          {}%
        }%
      }%
    }%
    {\glscustomtext\glsinsert}%
  }%
}
}
\newcommand*{\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%

```



```

name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%

```

```

\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
  \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*{\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[##1]{\glsentryfmt}%
}
\newcommand*{\DUANewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural

```

```

\let\org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@@gl@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
\let\gl@assign@descplural\org@gl@assign@descplural
}
\newcommand*{\SetDUASstyle}{%
  \renewcommand{\newacronym}[4][{}]{%
    \ifx\@gl@acronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDUADisplayStyle{\@gl@type}%
  }%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle
  \ifgl@acrdescription
    \ifgl@acrfootnote
      \SetDescriptionFootnoteAcronymStyle
    \else
      \ifgl@acrdua
        \SetDescriptionDUAAcronymStyle
      \else
        \SetDescriptionAcronymStyle
      \fi
    \fi
  \fi

```

```

\else
  \ifglacrfootnote
    \SetFootnoteAcronymStyle
  \else
    \ifthenelse{\boolean{glacrsmallcaps}\OR
      \boolean{glacrsmaller}}{%
      {%
        \SetSmallAcronymStyle
      }%
    }%
    \ifglacrdua
      \SetDUASstyle
    \fi
  }%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
  \defglentryfmt[#1]{\glsgenentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
  name={\the\glsshorttok},%
  description={\the\glslongtok},%
  first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
  firstplural={\acrfullformat
    {\noexpand\glsentrylongpl{\the\glslabeltok}}%
    {\noexpand\glsentryshortpl{\the\glslabeltok}}},%
  text={\the\glsshorttok},%
  plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      user1={\the\glsshorttok},%
      user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
      user3={\the\glslongtok},%
      user4={\the\glslongtok\noexpand\acrpluralsuffix},%
      \CustomAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}

```

```

\newcommand*{\SetCustomStyle}{%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@glsacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetCustomDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \CustomNewAcronymDef
  }%
  \@for\@gls@type:=\@glsacronymlists\do{%
    \SetCustomDisplayStyle{\@gls@type}%
  }%
}
\RequirePackage{glossary-hypernav}[=v4.46]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showglotype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
\expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}

```

```

\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@glstoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \csshow{glo@glstoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show@glsacronymlists
}
\newcommand*{\showglossaries}{%
  \show@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglsc-compatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.46]
\fi

```

8.2 Rollback v4.46 (glossaries-accsupp-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2020/03/19 v4.46 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}
{%
  \ifx@glsxtr@doaccsupp\empty
  \GlossariesWarning{The 'glossaries-accsupp'
    package has been loaded\MessageBreak
    after the 'glossaries-extra' package. This\MessageBreak
    can cause a failure to integrate both packages. \MessageBreak
    Either use the 'accsupp' option when you load\MessageBreak
    'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
    before loading 'glossaries-extra'}%

```

```

\fi
}
{}
\def\compatibleglossentry#1#2{%
  \toks@{#2}%
  \protected@edef\@do@glossentry{%
    \noexpand\accsuppglossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
  \toks@{#3}%
  \protected@edef\@do@subglossentry{%
    \noexpand\accsuppglossarysubentryfield{\number#1}%
    {#2}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
    {\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
    {\the\toks@}%
  }%
  \@do@subglossentry
}
\RequirePackage{glossaries}[v4.46]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
  \RequirePackage{accsupp}
  \renewcommand{\gls@accessibility}[4]{%
    \BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\glsaccessibility}[4][[]]{%
  \@glsshowaccsupp{#1}{#2}{#3}%
  \gls@accessibility{#1}{#2}{#3}{#4}%
}
\define@key{glossentry}{access}{%
  \def\@glo@access{#1}%
}
\define@key{glossentry}{textaccess}{%
  \def\@glo@textaccess{#1}%
}
\define@key{glossentry}{firstaccess}{%

```



```

\def\@glo@firstaccess{#1}%
}
\define@key{glossentry}{pluralaccess}{%
\def\@glo@pluralaccess{#1}%
}
\define@key{glossentry}{firstpluralaccess}{%
\def\@glo@firstpluralaccess{#1}%
}
\define@key{glossentry}{symbolaccess}{%
\def\@glo@symbolaccess{#1}%
}
\define@key{glossentry}{symbolpluralaccess}{%
\def\@glo@symbolpluralaccess{#1}%
}
\define@key{glossentry}{descriptionaccess}{%
\def\@glo@descaccess{#1}%
}
\define@key{glossentry}{descriptionpluralaccess}{%
\def\@glo@descpluralaccess{#1}%
}
\define@key{glossentry}{shortaccess}{%
\def\@glo@shortaccess{#1}%
}
\define@key{glossentry}{shortpluralaccess}{%
\def\@glo@shortpluralaccess{#1}%
}
\define@key{glossentry}{longaccess}{%
\def\@glo@longaccess{#1}%
}
\define@key{glossentry}{longpluralaccess}{%
\def\@glo@longpluralaccess{#1}%
}
\define@key{glossentry}{user1access}{%
\def\@glo@useriaccess{#1}%
}
\define@key{glossentry}{user2access}{%
\def\@glo@useriiaccess{#1}%
}
\define@key{glossentry}{user3access}{%
\def\@glo@useriiiaccess{#1}%
}
\define@key{glossentry}{user4access}{%
\def\@glo@userivaccess{#1}%
}
\define@key{glossentry}{user5access}{%
\def\@glo@uservaccess{#1}%
}
\define@key{glossentry}{user6access}{%
\def\@glo@userviaccess{#1}%
}
}

```

```

\appto\@gls@keymap{,%
  {access}{access},%
  {textaccess}{textaccess},%
  {firstaccess}{firstaccess},%
  {pluralaccess}{pluralaccess},%
  {firstpluralaccess}{firstpluralaccess},%
  {symbolaccess}{symbolaccess},%
  {symbolpluralaccess}{symbolpluralaccess},%
  {descaccess}{descaccess},%
  {descpluralaccess}{descpluralaccess},%
  {shortaccess}{shortaccess},%
  {shortpluralaccess}{shortpluralaccess},%
  {longaccess}{longaccess},%
  {longpluralaccess}{longpluralaccess},%
  {user1access}{useriaccess},%
  {user2access}{useriiaccess},%
  {user3access}{useriiiaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaccess{\relax}%
  \def\@glo@useriiiaccess{\relax}%
  \def\@glo@userivaccess{\relax}%
  \def\@glo@uservaccess{\relax}%
  \def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
  \@gls@oldnewglossaryentryposthook
  \expandafter
  \protected@xdef\csname glo@\@glo@label @access\endcsname{%

```

```

        \@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
    \@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
    \@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
    \@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
    \@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
    \@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
    \@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
    \@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
    \@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
    \@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
    \@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
    \@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
    \@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
    \@glo@useriaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiaaccess\endcsname{%
    \@glo@useriiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriiiaaccess\endcsname{%
    \@glo@useriiiaaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userivaccess\endcsname{%
    \@glo@userivaccess}%
\expandafter

```

```

\protected@xdef\csname glo@\@glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}
\newcommand*{\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*{\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}

```

```

}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glsxtr\glscategory{#4}#3accsupp}%
    {\csname glsxtr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glsxtr\glscategory{#4}accsupp}%
      {\csname glsxtr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}
\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
  #1%
  \else

```

```

        \xglaccsupp{\@glo@access}{#1}%
    \fi
}
\newcommand*{\@gls@fieldaccess@display}[4]{%
    \protected@edef\@glo@access{#4}%
    \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
    {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*{\@gls@nameaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@access}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{name}{#1}{\@glsentryaccess{#2}}}%
}
\newrobustcmd*{\@gls@textaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@textaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{text}{#1}{\@glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\@gls@pluralaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@pluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{plural}{#1}{\@glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@firstaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@firstaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{first}{#1}{\@glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\@gls@firstpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\@glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@symbolaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@symbolaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbol}{#1}{\@glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\@gls@symbolpluralaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@symbolpluralaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\@glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\@gls@descriptionaccessdisplay}[2]{%
    \ifcsundef{glo@\@glsdetoklabel{#2}@descaccess}%
    {#1}%
    {\@gls@fieldaccess@display{#2}{desc}{#1}{\@glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\@gls@descriptionpluralaccessdisplay}[2]{%

```

```

\ifcsundef{glo@glstdetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glstrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glstryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glstryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glstrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glstrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glseriaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@seriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{seri}{#1}{\glstryseriaccess{#2}}}%
}
\newrobustcmd*{\glseriiaaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@seriiaaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{seriia}{#1}{\glstryseriiaaccess{#2}}}%
}
\newrobustcmd*{\glseriiaaaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@seriiaaaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{seriiaa}{#1}{\glstryseriiaaaccess{#2}}}%
}
\newrobustcmd*{\glserivaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@serivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{seriv}{#1}{\glstryserivaccess{#2}}}%
}
\newrobustcmd*{\glserivaccessdisplay}[2]{%
\ifcsundef{glo@glstdetoklabel{#2}@serivaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{serv}{#1}{\glstryserivaccess{#2}}}%
}
\newrobustcmd*{\glserivaccessdisplay}[2]{%

```

```

\ifcsundef{glo@glstetoklabel{#2}@userviaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{uservi}{#1}{\glstentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
\ifcsundef{gls#1accessdisplay}%
{%
\PackageError{glossaries-accsupp}{No accessibility support
for key '#1'}{ }%
}%
{%
\csname gls#1accessdisplay\endcsname{#2}{#3}%
}%
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
\ifdefempty{glscustomtext}
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\glstentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glstentrydescplural{\glslabel}}{\glslabel}}%
{\glssymbolpluralaccessdisplay
{\glstentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glstentryfirstplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glstentrydescplural{\glslabel}}{\glslabel}}%
{\glssymbolpluralaccessdisplay
{\glstentrysymbolplural{\glslabel}}{\glslabel}}%
{\glsinsert}}%
}%
}%
}%
\ifglsused\glslabel
{%
#2{\glspluralaccessdisplay
{\Glsentryplural{\glslabel}}{\glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glstentrydescplural{\glslabel}}{\glslabel}}%
{\glssymbolpluralaccessdisplay
{\glstentrysymbolplural{\glslabel}}{\glslabel}}%
}

```



```

        {\glsinsert}%
    }%
    {%
        #1{\glsfirstpluralaccessdisplay
            {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
            {\glsdescriptionpluralaccessdisplay
                {\glsentrydescplural{\glslabel}}{\glslabel}}%
            {\glssymbolpluralaccessdisplay
                {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
            {\glsinsert}%
        }%
    }%
    {%
        \ifglsused\glslabel
        {%
            \MakeUppercase{%
                #2{\glspluralaccessdisplay
                    {\glsentryplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                        {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glssymbolpluralaccessdisplay
                        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
        {%
            \MakeUppercase{%
                #1{\glsfirstpluralaccessdisplay
                    {\glsentryfirstplural{\glslabel}}{\glslabel}}%
                    {\glsdescriptionpluralaccessdisplay
                        {\glsentrydescplural{\glslabel}}{\glslabel}}%
                    {\glssymbolpluralaccessdisplay
                        {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
                    {\glsinsert}}%
            }%
        }%
    }%
    {%
        \glscapscase
        {%
            \ifglsused\glslabel
            {%
                #2{\glstextaccessdisplay
                    {\glsentrytext{\glslabel}}{\glslabel}}%
                    {\glsdescriptionaccessdisplay
                        {\glsentrydesc{\glslabel}}{\glslabel}}%
                    {\glssymbolaccessdisplay
                        {\glsentrysymbol{\glslabel}}{\glslabel}}%
                    {\glsinsert}%
                }%
            }%
        }%
    }%

```

```

#1{\glsfirstaccessdisplay
  {\glsentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glsentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glsentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
  \ifglsused\glslabel
  {%
    #2{\glstextaccessdisplay
      {\Glsentrytext{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glssymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
    }%
  }%
  #1{\glsfirstaccessdisplay
    {\Glsentryfirst{\glslabel}}{\glslabel}}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glssymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \MakeUppercase{%
      #2{\glstextaccessdisplay
        {\glsentrytext{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
        {\glsinsert}}%
      }%
    }%
    \MakeUppercase{%
      #1{\glsfirstaccessdisplay
        {\glsentryfirst{\glslabel}}{\glslabel}}%
        {\glsdescriptionaccessdisplay
          {\glsentrydesc{\glslabel}}{\glslabel}}%
        {\glssymbolaccessdisplay
          {\glsentrysymbol{\glslabel}}{\glslabel}}%
      }%
    }%
  }%
}

```

```

        {\glsinsert}}}%
    }%
}%
}%
}%
{\%
\ifglsused{\glslabel}%
{\%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
{\%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
     {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glsymbolaccessdisplay
     {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
}%
}%
}
\renewcommand*{\glsгенentryfmt}{\%
\ifdefempty\glscustomtext
{\%
    \glsifplural
    {\%
        \glscapscase
        {\%
            \ifglsused\glslabel
            {\%
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
            {\%
                \glsfirstpluralaccessdisplay
                {\glsentryfirstplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
        }%
        {\%
            \ifglsused\glslabel
            {\%
                \glspluralaccessdisplay
                {\glsentryplural{\glslabel}}{\glslabel}%
                \glsinsert
            }%
        }%
    }%
}

```

```

}%
{%
  \glsfirstpluralaccessdisplay
    {\Glsentryfirstplural{\glslabel}}{\glslabel}%
  \glsinsert
}%
}%
{%
  \ifglsused\glslabel
  {%
    \glspluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentryplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
  {%
    \glsfirstpluralacessdisplay
      {\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
      {\glslabel}%
    \mfirstucMakeUppercase{\glsinsert}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%
      \glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
      \glsinsert
    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglsused\glslabel
    {%

```

```

        \glstextaccessdisplay
        {\mfirstucMakeUppercase{\glentrytext{\glslabel}}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
    }%
    {%
        \glsfirstaccessdisplay
        {\mfirstucMakeUppercase{\glentryfirst{\glslabel}}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
    }%
}%
}%
}%
{%
    \glscustomtext\glsinsert
}%
}
\renewcommand*{\glsgenacfmt}{%
    \ifdefempty\glscustomtext
    {%
        \ifglused\glslabel
        {%
            \glsifplural
            {%
                \glscapscase
                {%
                    \acronymfont
                    {\glsshortpluralaccessdisplay
                     {\glentryshortpl{\glslabel}}{\glslabel}}%
                    \glsinsert
                }%
            }%
            {%
                \acronymfont
                {\glsshortpluralaccessdisplay
                 {\Glsentryshortpl{\glslabel}}{\glslabel}}%
                \glsinsert
            }%
            {%
                \mfirstucMakeUppercase
                {\acronymfont
                 {\glsshortpluralaccessdisplay
                  {\glentryshortpl{\glslabel}}{\glslabel}}%
                 \glsinsert}%
            }%
        }%
    }%
    {%
        \glscapscase
        {%
            \acronymfont
            {\glsshortaccessdisplay{\glentryshort{\glslabel}}{\glslabel}}%
            \glsinsert
        }%
    }%
}

```

```

    }%
    {%
      \acronymfont
      {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
      \glsinsert
    }%
    {%
      \mfirstucMakeUppercase
      {\acronymfont{%
        \glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert}%
    }%
  }%
}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
  {%
    \glscapscase
    {%
      \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \mfirstucMakeUppercase
      {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}%
}%
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space

```

```

(\glsshortaccessdisplay{\protect\firstacronymfont{\glentryshort{#1}}}{#1})%
}
\renewcommand*\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\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@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glentryshort{#2}}{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree

```

```

\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glsshortaccessdisplay
    {\MakeUppercase{\glsentryshort{#2}}}{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@AcrLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@ACRLong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glslongaccessdisplay{%
        \MakeUppercase{\glsentrylong{#2}}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%

```



```

\glspostlinkhook
}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\glstrytext{#2}}{#2}#3}%
}
\def\@GLstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glstextaccessdisplay{\Glsentrytext{#2}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glstextaccessdisplay{\mfirstucMakeUppercase{\glstrytext{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\glstryfirst{#2}}{#2}#3}%
}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstaccessdisplay{\Glsentryfirst{#2}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstaccessdisplay{\mfirstucMakeUppercase{\glstryfirst{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\glstryplural{#2}}{#2}#3}%
}
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glspluralaccessdisplay{\Glsentryplural{#2}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glspluralaccessdisplay{\mfirstucMakeUppercase{\glstryplural{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\glstryfirstplural{#2}}{#2}#3}%
}
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsfirstpluralaccessdisplay{\Glsentryfirstplural{#2}}{#2}#3}%
}
\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glstryfirstplural{#2}}{#2}%
      \mfirstucMakeUppercase{#3}}}%
  }
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glstryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsnameaccessdisplay{\mfirstucMakeUppercase{\Glsentryname{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\Glsentrydesc{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\Glsentrydescplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolaccessdisplay{\mfirstucMakeUppercase{\Glsentrysymbol{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsymbolpluralaccessdisplay{\Glsentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}%
{\glsymbolpluralaccessdisplay{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}

```

```

}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glsentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glsentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glsentryuserii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@glsuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glsentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glsentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriii{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@glsuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glsentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glsentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}

```

```

}
\def\@GLSuser@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseraccessdisplay{\mfirstucMakeUppercase{\glsentryuser{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glsentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glsentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glsentryuseri{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\Glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%

```

```

        \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
    }%
}
\newcommand*{\accsuppglossaryentryfield}[5]{%
    \glossaryentryfield{#1}%
    {\glanameaccessdisplay{#2}{#1}}%
    {\gldescriptionaccessdisplay{#3}{#1}}%
    {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*{\accsuppglossarysubentryfield}[6]{%
    \glossarysubentryfield{#1}{#2}%
    {\glanameaccessdisplay{#3}{#2}}%
    {\gldescriptionaccessdisplay{#4}{#2}}%
    {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
    \renewcommand*{\genacrfullformat}[2]{%
        \glslongaccessdisplay{\glentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glentryshort{##1}}}{##1})%
    }%
    \renewcommand*{\Genacrfullformat}[2]{%
        \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
        (\glsshortaccessdisplay
            {\protect\firstacronymfont{\glentryshort{##1}}}{##1})%
    }%
    \renewcommand*{\genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*{\Genplacrfullformat}[2]{%
        \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
        (\glsshortpluralaccessdisplay
            {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1})%
    }%
    \renewcommand*{\acronymentry}[1]{%
        \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}
    }
    \renewcommand*{\acronymsort}[2]{##1}%
    \renewcommand*{\acronymfont}[1]{##1}%
    \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
    \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{short-long}%
{%

```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\Glsentryshort{##1}}}{##1}##2\space
(\glslongaccessdisplay{\glsentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay
{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2\space
(\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}})%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
          \mfirstucMakeUppercase{\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
      }%
      {%
        \glslongaccessdisplay
          {\mfirstucMakeUppercase
            {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
        \mfirstucMakeUppercase{\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%

```



```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}})%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}})%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}})%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\Glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2}})%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
    {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
    (\glsshortpluralaccessdisplay
    {\acronymfont{\glsentryshortpl{##2}}}{##2}})%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
  (\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%

```

```

\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
  \GlsUseAcrEntryDispStyle{dua}%
}%
{%
  \GlsUseAcrStyleDefs{dua}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}}%
  \renewcommand*{\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \glshyperfirstfalse
  \renewcommand*{\genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
    \protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glsshortaccessdisplay
      {\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
    \protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}}{##1}}%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
    \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
    \protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}}{##1}}%
  }%
  \renewcommand*{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2}##3\space
    }
  }

```

```

        (\glslongaccessdisplay{\glsentrylong{##2}}{##2}})%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##2}}}{##2}##3\space
        (\glslongaccessdisplay{\glsentrylong{##2}}{##2}})%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortaccessdisplay
            {\mfirstucMakeUppercase
                {\acronymfont{\glsentryshort{##2}}}{##2}##3\space
                (\glslongaccessdisplay{\glsentrylong{##2}}{##2}})}}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
            (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\acronymfont{\Glsentryshortpl{##2}}}{##2}##3\space
            (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})%
\renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \glsshortpluralaccessdisplay
            {\mfirstucMakeUppercase
                {\acronymfont{\glsentryshortpl{##2}}}{##2}##3\space
                (\glslongpluralaccessdisplay{\glsentrylongpl{##2}}{##2}})}}%
\renewcommand*{\glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1}})%
\renewcommand*{\Glsentryfull}[1]{%
    \glsshortaccessdisplay{\acronymfont{\Glsentryshort{##1}}}{##1}\space
    (\glslongaccessdisplay{\glsentrylong{##1}}{##1}})%
\renewcommand*{\glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##1}}}{##1}\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}})%
\renewcommand*{\Glsentryfullpl}[1]{%
    \glsshortpluralaccessdisplay
        {\acronymfont{\Glsentryshortpl{##1}}}{##1}\space
        (\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}})%
}
\renewacronymstyle{footnote-sc}%
{%
    \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
    \GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}

```

```

\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDispStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}

```

```

\newcommand*{\glsdefaultshortaccess}[2]{#1}
\renewcommand*{\newacronymhook}{%
  \edef\@gls@keylist{%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    \the\glskeylisttok}%
  \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
}
\renewcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      description={\the\glslongtok},%
      descriptionaccess=\relax,%
      text={\the\glsshorttok},%
      access={\noexpand\@glo@textaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      first={\noexpand\glslongaccessdisplay
        {\the\glslongtok}{\the\glslabeltok}\space
        (\noexpand\glsshortaccessdisplay
          {\the\glsshorttok}{\the\glslabeltok})},%
      plural={\the\glsshorttok\acrpluralsuffix},%
      firstplural={\noexpand\glslongpluralaccessdisplay
        {\noexpand\@glo@longpl}{\the\glslabeltok}\space
        (\noexpand\glsshortpluralaccessdisplay
          {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
      firstaccess=\relax,%
      firstpluralaccess=\relax,%
      textaccess={\noexpand\@glo@shortaccess},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%

```

```

    @@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
text={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
access={\noexpand\@glo@textaccess},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\the\glslongtok},%
symbolplural={\the\glslongtok\noexpand\acrpluralsuffix},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%

```

```

type=\acronymtype,%
name={\noexpand
  \acronymformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\@glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolaccess={\noexpand\@glo@textaccess},%
symbolplural={\noexpand\@glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\@glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\renewcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    textaccess={\the\glslongtok},%
    access={\noexpand\@glo@textaccess},%

```

```

plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\the\glslongtok},%
descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gls@assign@plural\gls@assign@plural
\let\org@gls@assign@firstpl\gls@assign@firstpl
\let\org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\@do@newglossaryentry
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@descplural\org@gls@assign@descplural
}
\renewcommand*\SmallNewAcronymDef{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    access={\noexpand\@glo@symbolaccess},%
    sort={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    text={\noexpand\@glo@short},%
    textaccess={\noexpand\@glo@shortaccess},%
    plural={\noexpand\@glo@shortpl},%
    first={\the\glslongtok},%
    firstaccess=\relax,
    firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\noexpand\@glo@first},%
    descriptionplural={\noexpand\@glo@firstplural},%
    symbol={\the\glsshorttok},%
    symbolaccess={\the\glslongtok},%

```



```

symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\org@gl@assign@firstpl\gl@assign@firstpl
\let\org@gl@assign@plural\gl@assign@plural
\let\org@gl@assign@descplural\gl@assign@descplural
\let\org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\org@gl@assign@firstpl
\let\gl@assign@plural\org@gl@assign@plural
\let\gl@assign@descplural\org@gl@assign@descplural
\let\gl@assign@symbolplural\org@gl@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglonameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglofirstpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglofirstplurallaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstplurallaccess\endcsname
}
\newcommand*{\showglosymbolaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglosymbolpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}

```

```

}
\newcommand*{\showglodescaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*{\showglodescpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*{\showgloshortaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*{\showgloshortpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*{\showglolongaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*{\showglolongpluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}

```

8.3 Rollback v4.46 (glossaries-babel-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
  }%
}
}

```

8.4 Rollback v4.46 (glossaries-compatible-207-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2020/03/19 v4.46 (NLCT)]
\ifglxsindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J}%
    (markup-locref
    :open \string"\string~\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter\@gobble\string\{\string" ^^J
    :close \string"\expandafter\@gobble\string\}\string" ^^J
    :attr \string"#1\string"))}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\ifglxsindy
  \renewcommand*\{GlsAddXdyLocation}[2]{%
    \edef\@xdyuserlocationdefs{%
      \@xdyuserlocationdefs ^^J%
      (define-location-class \string"#1\string"^^J\space\space
      \space(#2))

```

```

}%
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#1\string"}%
}
\fi
\renewcommand{\@do@wrglossary}[1]{%
\ifglxsindy
  \expandafter\@glo@check@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix\relax
    \def\@glo@range{:close-range}%
  \fi
\fi
\protected@edef\@glslocref{\theglsentrycounter}%
\@gls@checkmkidxchars\@glslocref
\glossary[\csname glo@#1@type\endcsname]{%
(indexentry :tkey (\csname glo@#1@index\endcsname)
:locref \string"\@glslocref\string" %
:attr \string"\@glo@suffix\string" \@glo@range
)
}%
\else
  \set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
  \glossary[\csname glo@#1@type\endcsname]{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar\@glo@numfmt}{\theglsentrycounter}}%
\fi
}
\def\@set@glo@numformat#1#2#3{%
  \expandafter\@glo@check@mkidxrangechar#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[] {#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}
\ifglxsindy
\def\writeist{%
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty

```

```

\else
\protected@write\glswrite{}\{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@tmp{\string" \string"roman-numbers-lowercase\string"
:sep \string"}%
\@onelevel@sanitize\@tmp
\ifx\@tmp\@gls@roman
\write\glswrite{(define-location-class
\string"roman-page-numbers\string"^^J\space\space\space
(\string"roman-numbers-lowercase\string")
:min-range-length \@glsminrange)}%
\else
\write\glswrite{(define-location-class
\string"roman-page-numbers\string"^^J\space\space\space
(:sep "\@gls@roman")
:min-range-length \@glsminrange)}%
\fi
\write\glswrite{(define-location-class
\string"Roman-page-numbers\string"^^J\space\space\space
(\string"roman-numbers-uppercase\string")
:min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
\string"arabic-page-numbers\string"^^J\space\space\space
(\string"arabic-numbers\string")
:min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
\string"alpha-page-numbers\string"^^J\space\space\space
(\string"alpha\string")
:min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
\string"Alpha-page-numbers\string"^^J\space\space\space
(\string"ALPHA\string")
:min-range-length \@glsminrange)}%
\write\glswrite{(define-location-class
\string"Appendix-page-numbers\string"^^J\space\space\space
(\string"ALPHA\string"
:sep \string"\@glsAlphacompositor\string"
\string"arabic-numbers\string")
:min-range-length \@glsminrange)}%

```

```

\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
    :sep \string"\glscpositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmrange)}}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\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~n\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsoopenbrace\string\glossaryentrynumbers
  \glsoopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list
  :sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
  :sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffiF
\@onelevel@sanitize\gls@suffiFF
\ifx\gls@suffiF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffiF" :length 1 :ignore-end)}%
\fi

```

```

\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{(markup-range
    :close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
  :sep \string\string\glsgroupskip\string~n\string)}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
  :open-head \string\string\glsgroupheading
  \glsopenbrace\string^^J\space\space\space
  :close-head \string\glsclosebrace\string)}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
  \openout\glswrite=\istfilename
  \write\glswrite{\expandafter\@gobble\string\% makeindex style file
    created by the glossaries package}
  \write\glswrite{\expandafter\@gobble\string\% for document
    '\jobname' on \the\year-\the\month-\the\day}
  \write\glswrite{actual '\@gls@actualchar'}
  \write\glswrite{encap '\@gls@encapchar'}
  \write\glswrite{level '\@gls@levelchar'}
  \write\glswrite{quote '\@gls@quotechar'}
  \write\glswrite{keyword \string\string\glossaryentry\string"}
  \write\glswrite{preamble \string\string\glossarysection[\string
    \glossarytoctitle]{\string\glossarytitle}\string
    \glossarypreamble\string~n\string\begin{theglossary}\string
    \glossaryheader\string~n\string"}
  \write\glswrite{postamble \string\string\string~n\string
    \end{theglossary}\string\glossarypostamble\string~n
    \string"}
  \write\glswrite{group_skip \string\string\glsgroupskip\string~n
    \string"}
  \write\glswrite{item_0 \string\string\string~n\string"}
  \write\glswrite{item_1 \string\string\string~n\string"}
  \write\glswrite{item_2 \string\string\string~n\string"}
  \write\glswrite{item_01 \string\string\string~n\string"}
  \write\glswrite{item_x1

```

```

\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
\string"\string\relax \string\glsresetentrylist\string\n
\string"}
\write\glswrite{delim_0 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
\glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string\}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
\string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
\string"\string\}\string\relax
\string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glssymbols\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.5 Rollback v4.46 (glossaries-compatible-307-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2020/03/19 v4.46 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
\ifcsundef{@glscompstyle@#1}%
{%
\csdef{@glscompstyle@#1}{#2}%
}%
}

```



```

    {%
      \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
    }%
  }
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstreempty{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
        {%
          \glsresetsubentrycounter
          \glsinlineparentchildseparator
          \def\gls@inlinesubsep{}%
          \def\gls@inlinepostchild{\glsinlinepostchild}%
        }%
        {}%
      \def\gls@inlinesep{\glsinlineseparator}%
    }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
      ##3\glspostdescription\space ##5%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
  }
\compatglossarystyle{listgroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
  \csuse{@glscompstyle@list}%
}%
\compatglossarystyle{altlist}{%

```

```

\renewcommand*{\glossaryentryfield}[5]{%
  \item[\glentryitem{##1}\glstarget{##1}{##2}]%
  \mbox{}\par\nobreak\@afterheading
  ##3\glspostdescription\space ##5}%
\renewcommand{\glossarysubentryfield}[6]{%
  \par
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{altlistgroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
  \csuse{@glscompstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}\glstarget{##1}{##2}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \item[]\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{##3}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
  }%
\compatglossarystyle{sublistdotted}{%
  \csuse{@glscompstyle@listdotted}%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item[\glentryitem{##1}\glstarget{##1}{##2}]}%
}%
\compatglossarystyle{long}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
  }%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%

```

```

\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
\csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
\csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
        \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%

```

```

\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5}%
\renewcommand*\glossarysubentryfield}[6]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6}%
}%
\compatglossarystyle{indexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%

```

```

\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{alttree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else
\settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
\hangindent\glstreeindent
\parindent\glstreeindent
\fi
\makebox[Opt][r]{\makebox[\glstreeindent][l]{%
\glstentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
\ifx\relax##4\relax
\else
(##4)\space
\fi
##3\glspostdescription \space ##5\par
\def\@gls@prevlevel{0}%

```

```

}%
\renewcommand{\glossarysubentryfield}[6]{%
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \ifundefined{\@glswidestname\romannumeral##1}{%
      \settowidth{\@gls@tmplen}{\textbf{\@glswidestname\space}}{%
      \settowidth{\@gls@tmplen}{\textbf{%
        \csname \@glswidestname\romannumeral##1\endcsname\space}}}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\@gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \ifundefined{\@glswidestname\romannumeral\@gls@prevlevel}{%
        \settowidth{\glstreeindent}{\textbf{%
          \@glswidestname\space}}}%
        \settowidth{\glstreeindent}{\textbf{%
          \csname \@glswidestname\romannumeral\@gls@prevlevel
            \endcsname\space}}}%
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
      \fi
    \fi
    \hangindent\glstreeindent
    \makebox[0pt][r]{\makebox[\@gls@tmplen][l]{%
      \textbf{\glstarget{##2}{##3}}}%
    \ifx##5\relax\relax
    \else
      (##5)\space
    \fi
    ##4\glspostdescription\space ##6\par
    \def\@gls@prevlevel{##1}%
  }%
}%
\compatglossarystyle{almtreegroup}{%
  \csuse{\@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
  \csuse{\@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
  \csuse{\@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
  \csuse{\@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%

```

```

\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenoname}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenonamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glsentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%

```



```

\renewcommand*{\glossaryentryfield}[5]{%
  \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
  &
  \glssubentryitem{##2}%
  \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
  \csuse{@glscmpstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
  \csuse{@glscmpstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%

```

```

\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscompstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
  \csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
  \csuse{@glscompstyle@super4col}%
}%

```

8.6 Rollback v4.46 (glossaries-polyglossia-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \PackageWarningNoLine{glossaries}%
      {No language module detected for ‘\this@dialect’.\MessageBreak
      Language modules need to be installed separately.\MessageBreak
      Please check on CTAN for a bundle called\MessageBreak
      ‘glossaries-\CurrentTrackedLanguage’ or similar}%
    }%
  }%
}%
{}%
```

8.7 Rollback v4.46 (glossaries-prefix-2020-03-19.sty)

Version 4.46 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2020/03/19 v4.46 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[=v4.46]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%
  {prefixfirst}{prefixfirst},%
  {prefixfirstplural}{prefixfirstplural},%
  {prefix}{prefix},%
  {prefixplural}{prefixplural}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
```

```

\let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
\gls@assign@field{}\@glo@label\{prefix\}\@glo@entryprefix}%
\gls@assign@field{}\@glo@label\{prefixplural\}\@glo@entryprefixplural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefix\endcsname\}\@glo@label\{prefixfirst}%
{\@glo@entryprefixfirst}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @prefixplural\endcsname\}\@glo@label}%
{prefixfirstplural}\@glo@entryprefixfirstplural}%
}
\newcommand*\@glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*\@glsentryprefixfirstplural}[1]{%
\csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*\@glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*\@glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*\@Glsentryprefixfirst}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixfirstplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefix}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
\xmakefirstuc\@glo@text
}
\newrobustcmd*\@Glsentryprefixplural}[1]{%
\protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
\xmakefirstuc\@glo@text
}
\newcommand*\@ifglshasprefix}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefix}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixplural}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixplural}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirst}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixfirst}%
{#3}%
{#2}%
}
\newcommand*\@ifglshasprefixfirstplural}[3]{%
\ifcsempy{glo@\glsdetoklabel{#1}@prefixfirstplural}%

```

```

    {#3}%
    {#2}%
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
    {\@glsxtr@record{#1}{#2}{glslink}}%
  }%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@gls@hyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
    {\@pgls@{#1}{#2}}%
    {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
    \@gls@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@gls@hyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
    {\@pglsp1@{#1}{#2}}%
    {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifgls@hasprefixplural{#2}{\glsentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
      \ifgls@hasprefixfirstplural{#2}%
      {\glsentryprefixfirstplural{#2}\glsprefixsep}{}%
    }%
    \@glspl@{#1}{#2}[#3]%
  }%
}

```

```

\newrobustcmd{\Pgl}{\@gls@hyp@opt\Pgl}
\newcommand*{\@Pgl}[2][{}]{%
  \new@ifnextchar[%
    {\@Pgl@{#1}{#2}}%
    {\@Pgl@{#1}{#2}[]}%
}
\def\@Pgl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefix{#2}%
      {%
        \Glsentryprefix{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {%
        \Glsentryprefixfirst{#2}%
        \glsprefixsep
        \@gls@{#1}{#2}[#3]%
      }%
      {\@Gls@{#1}{#2}[#3]}%
    }%
  }%
}
\newrobustcmd{\Pglsp}{\@gls@hyp@opt\Pglsp}
\newcommand*{\@Pglsp}[2][{}]{%
  \new@ifnextchar[%
    {\@Pglsp@{#1}{#2}}%
    {\@Pglsp@{#1}{#2}[]}%
}
\def\@Pglsp@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsdoifexists{#2}%
  {%
    \ifglsused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {%
        \Glsentryprefixplural{#2}%
        \glsprefixsep
        \@glspl@{#1}{#2}[#3]%
      }%
      {\@Glspl@{#1}{#2}[#3]}%
    }%
  }%
}

```

```

}%
{%
  \ifglshasprefixfirstplural{#2}%
  {%
    \Glsentryprefixfirstplural{#2}%
    \glsprefixsep
    \@glsp1@{#1}{#2}[#3]%
  }%
  {\@Glspl@{#1}{#2}[#3]}%
}%
}%
}
\newrobustcmd{\PGLS}{\@glshyp@opt\PGLS}
\newcommand*{\@PGLS}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLS@{#1}{#2}}%
    {\@PGLS@{#1}{#2}[]}%
}
\def\@PGLS@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSpl}{\@glshyp@opt\PGLSpl}
\newcommand*{\@PGLSpl}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLSpl@{#1}{#2}}%
    {\@PGLSpl@{#1}{#2}[]}%
}
\def\@PGLSpl@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glstoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefixplural{#2}%
      {\mfirstucMakeUppercase{\glentryprefixplural{#2}\glsprefixsep}}{%
    }%
    {%

```

```

\ifglshasprefixfirstplural{#2}%
{\mfirstucMakeUppercase{\glstryprefixfirstplural{#2}\glsprefixsep}}{}%
}%
\@GLSpl@{#1}{#2}{#3}%
}%
}

```

8.8 Rollback v4.46 (glossary-hypernav-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2020/03/19 v4.46 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’~Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@prerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
    \fi
  }
  \let\gls@hypergroup@prerun\relax
  \AtEndDocument{\gls@hypergroup@prerun}
  \newcommand*{\@gls@hypergroup}[2]{%
    \ifundefined{\@gls@hypergroup@list@#1}{%
      \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
    }{%
      \expandafter\let\expandafter\@gls@tmp
        \csname @gls@hypergroup@list@#1\endcsname
      \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{%
        \@gls@tmp,#2}%
    }%
  }
  }
  \newcommand*{\glsnavigation}{%
    \def\@gls@between{}%
    \ifcsundef{\@gls@hypergroup@list@\@glo@type}%

```



```

{%
  \def\@gls@list{%
}%
{%
  \expandafter\let\expandafter\@gls@list
    \csname @gls@hypergroup@list@\@glo@type\endcsname
}%
\@for\@gls@tmp:=\@gls@list\do{%
  \@gls@between
  \@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
  \glsnavhyperlink{\@gls@tmp}{\@gls@grptitle}%
  \let\@gls@between\glshypernavsep
}%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glssymbolnav}{%
  \glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
  \glshypernavsep
  \glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
  \glshypernavsep
}

```

8.9 Rollback v4.46 (glossary-inline-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2020/03/19 v4.46 (NLCT)]
\newglossarystyle{inline}{%
  \renewenvironment{theglossary}%
  {%
    \def\gls@inlinesep{%
    \def\gls@inlinesubsep{%
    \def\gls@inlinepostchild{%
    }%
    {\glspostinline}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{%
  \renewcommand{\glossentry}[2]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \glsentryitem{##1}%
    \glsinlinenameformat{##1}%
    \glossentryname{##1}%
  }%
  \ifglsdescsuppressed{##1}%
  {%
    \glsinlineemptydescformat
    {%
      \glossentrysymbol{##1}%
    }%
  }%
}

```

```

    {%
      ##2%
    }%
  }%
  {%
    \ifglshasdesc{##1}%
    {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
    {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
  }%
  \ifglshaschildren{##1}%
  {%
    \glsresetsubentrycounter
    \glsinlineparentchildseparator
    \def\gls@inlinesubsep{}%
    \def\gls@inlinepostchild{\glsinlinepostchild}%
  }%
  {}%
  \def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glssubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{\}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{\}
\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{\}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.10 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-list}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%

```

```

\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*\glslistgroupheaderfmt[1]{#1}
\newcommand*\glslistnavigationitem[1]{\item[#1]}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}%
    {\begin{description}}{\end{description}}%
  \renewcommand*\glossaryheader{}%
  \renewcommand*\glsgroupheading[1]{}%
  \renewcommand*\glossentry[2]{%
    \item[\glstryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}]
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*\glsgroupskip{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glslnavigation}}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt
      {\glslnavhypertarget{##1}{\glsgrouptitle{##1}}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*\glossentry[2]{%
    \item[\glstryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}]
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*\subglossentry[3]{%
    \par
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
}
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glsgroupheading[1]{%
    \item[\glslistgroupheaderfmt{\glsgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*\glossaryheader{%
    \glslistnavigationitem{\glslnavigation}}%

```

```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}]%
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
  \renewcommand*{\subglossentry}[3]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glssubentryitem{##2}%
      \glstarget{##2}{\glossentryname{##2}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
  }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glentryitem{##1}\glstarget{##1}{\glossentryname{##1}}]}%
}

```

8.11 Rollback v4.46 (glossary-list-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength\glsdescwidth
  \setlength{\glsdescwidth}{0.6\hsize}
}{%
}
\@ifundefined{glspagelistwidth}{%
  \newlength\glspagelistwidth
  \setlength{\glspagelistwidth}{0.1\hsize}
}{%
}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{\glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand*{\subglossentry}[3]{%
    &
  }

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
    \fi
}
\newglossarystyle{longborder}{%
    \setglossarystyle{long}%
    \renewenvironment{theglossary}{%
        \begin{longtable}{|l|p{\glstdescwidth}|}{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longheader}{%
    \setglossarystyle{long}%
    \renewcommand*{\glossaryheader}{%
        \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
}
\newglossarystyle{longheaderborder}{%
    \setglossarystyle{longborder}%
    \renewcommand*{\glossaryheader}{%
        \hline\bfseries \entryname & \bfseries
        \descriptionname\tabularnewline\hline
        \endhead
        \hline\endfoot}%
}
\newglossarystyle{long3col}{%
    \renewenvironment{theglossary}{%
        \begin{longtable}{lp{\glstdescwidth}p{\glspagelistwidth}}%
        {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupeheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glssubentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
    \fi

```

```

}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|p{\glstdescwidth}|p{\glspagelistwidth}|}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l|l|l|l}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&

```

```

        \bfseries \symbolname&
        \bfseries\pagelistname\tabularnewline\endhead}%
    }
    \newglossarystyle{long4colborder}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|l|l|l|l|}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
    }
    \newglossarystyle{long4colheaderborder}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|l|l|l|l|}}%
            {\end{longtable}}%
        \renewcommand*{\glossaryheader}{%
            \hline\bfseries\entryname&\bfseries\descriptionname&
            \bfseries \symbolname&
            \bfseries\pagelistname\tabularnewline\hline\endhead
            \hline\endfoot}%
    }
    \newglossarystyle{altlong4col}{%
        \setglossarystyle{long4col}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colheader}{%
        \setglossarystyle{long4colheader}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{lp{\glsdescwidth}lp{\glspagelistwidth}}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colborder}{%
        \setglossarystyle{long4colborder}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
            {\end{longtable}}%
    }
    \newglossarystyle{altlong4colheaderborder}{%
        \setglossarystyle{long4colheaderborder}%
        \renewenvironment{theglossary}%
            {\begin{longtable}{|lp{\glsdescwidth}|lp{\glspagelistwidth}|}}%
            {\end{longtable}}%
    }
}

```

8.12 Rollback v4.46 (glossary-longbooktabs-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname &
    \bfseries \descriptionname &
    \bfseries \symbolname &
    \bfseries \pagelistname
    \tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{altlong4col-booktabs}{%
  \glspatchLToutput

```



```

\setglossarystyle{long4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{altlong4col-booktabs}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}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@open
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@ \pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
          \fi
          \outputpage
        \setbox\z@\vbox{\box\LT@head\glslTpenaltycheck}%
      \fi
    }
  }

```

```

        \fi
    \fi
    \global\@colroom\@colht
    \global\vsizel\@colht
    {\unvbox\z@ \box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
\fi
\else
    \setbox\@ccclv\vbox{\unvbox\@ccclv\copy\LT@foot\vss}%
    \@makecol
    \@outputpage
    \global\vsizel\@colroom
    \copy\LT@head
    \glstpenaltycheck
    \nobreak
\fi
}%
}

```

8.13 Rollback v4.46 (glossary-longragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-longragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
    \newlength\glsdescwidth
    \setlength{\glsdescwidth}{0.6\hspace}
}{%
\@ifundefined{glspagelistwidth}{%
    \newlength\glspagelistwidth
    \setlength{\glspagelistwidth}{0.1\hspace}
}{%
\newglossarystyle{longragged}{%
    \renewenvironment{theglossary}%
        {\begin{longtable}[1>\raggedright]p{\glsdescwidth}}}%
        {\end{longtable}}}%
    \renewcommand*\glossaryheader{}%
    \renewcommand*\glsgroupheading[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1}\glspostdescription\space ##2%
        \tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2}%
        \glspostdescription\space ##3%
        \tabularnewline
    }%
}

```

```

}%
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
\fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
  }
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
      \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
      \glossentrydesc{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2} &
      ##3\tabularnewline
    }%
  }
\ifglsgroupskip
  \renewcommand*{\glsgroupskip}{}%
\else
  \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|}%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{altlongragged4colheader}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%

```

```

        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}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{\glsgdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
  \setglossarystyle{altlongragged4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|>{\raggedright}p{\glsgdescwidth}|l|%
      >{\raggedright}p{\glspagelistwidth}|}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

8.14 Rollback v4.46 (glossary-mcols-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.46]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }
  }

```

```

    }%
    {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcoltree}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
}

```

```

\renewcommand*{\glsgroupeheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
\renewcommand*{\glsgroupeheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand{\glsgroupeheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
\setglossarystyle{mcoltreenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupeheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnahypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%

```

```

}%
{\end{multicols}}}%
\renewcommand*{\glsgroupheading}[1]{%
  \par\noindent
  \glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
  \indexspace}%
}
\newglossarystyle{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalttreegroup}{%
  \setglossarystyle{mcolalttree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalttreehypergroup}{%
  \setglossarystyle{mcolalttree}%
  \renewcommand*{\glossaryheader}{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par
    \def\@gls@prevlevel{-1}%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstreegroupheaderfmt{\glsnavigationtarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalttreespannav}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%

```



```

\renewcommand*{\glsgroupheading}[1]{%
  \par
  \def\@gls@prevlevel{-1}%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
  \indexspace}%
}

```

8.15 Rollback v4.46 (glossary-super-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}\%
     \begin{supertabular}{lp{glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%

```

```

        \begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheader}{%
        \setglossarystyle{super}%
        \renewenvironment{theglossary}%
        {\tablehead{\bfseries \entryname &
            \bfseries\descriptionname\tabularnewline}%
            \tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}}}%
        {\end{supertabular}}%
    }
    \newglossarystyle{superheaderborder}{%
        \setglossarystyle{super}%
        \renewenvironment{theglossary}%
        {\tablehead{\hline\bfseries \entryname &
            \bfseries \descriptionname\tabularnewline\hline}%
            \tabletail{\hline}}
        {\begin{supertabular}{|l|p{\glsdescwidth}|}%
        {\end{supertabular}}%
    }
    \newglossarystyle{super3col}{%
        \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}}%
        {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
        {\end{supertabular}}%
        \renewcommand*{\glossaryheader}{}%
        \renewcommand*{\glsgroupheading}[1]{}%
        \renewcommand{\glossentry}[2]{%
            \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
            \glossentrydesc{##1} & ##2\tabularnewline
        }%
        \renewcommand{\subglossentry}[3]{%
            &
            \glssubentryitem{##2}%
            \glstarget{##2}{\strut}\glossentrydesc{##2} &
            ##3\tabularnewline
        }%
        \ifglsgroupskip
            \renewcommand*{\glsgroupskip}{}%
        \else
            \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
        \fi
    }
    \newglossarystyle{super3colborder}{%
        \setglossarystyle{super3col}%
        \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
            \begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}%
            {\end{supertabular}}%
    }

```

```

}
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsgdescwidth}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
    {\tabletail{\hline}%
      \begin{supertabular}{|l|p{\glsgdescwidth}|p{\glspagelistwidth}|}%
      {\end{supertabular}}}%
    }%
}
\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{llll}}{%
      \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \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}{l|l|l}}%
        {\end{supertabular}}%
    }
    \newglossarystyle{super4colborder}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline}\tabletail{\hline}%
             \begin{supertabular}{l|l|l|l|l}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{super4colheaderborder}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
                        \bfseries\symbolname &
                        \bfseries\pagelistname\tabularnewline\hline}%
             \tabletail{\hline}%
             \begin{supertabular}{l|l|l|l|l}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4col}{%
        \setglossarystyle{super4col}%
        \renewenvironment{theglossary}%
            {\tablehead{} \tabletail{}}%
            {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colheader}{%
        \setglossarystyle{super4colheader}%
        \renewenvironment{theglossary}%
            {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
                        \bfseries\symbolname &
                        \bfseries\pagelistname\tabularnewline}\tabletail{}}%
            {\begin{supertabular}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colborder}{%
        \setglossarystyle{super4colborder}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline}\tabletail{\hline}%
             \begin{supertabular}%
                 {l|lp{\glsgdescwidth}|l|lp{\glspagelistwidth}|}}%
            {\end{supertabular}}%
    }
    \newglossarystyle{altsuper4colheaderborder}{%
        \setglossarystyle{super4colheaderborder}%
        \renewenvironment{theglossary}%
            {\tablehead{\hline
                        \bfseries\entryname &
                        \bfseries\descriptionname &

```

```

        \bfseries\symbolname &
        \bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}%
  {|l|p{\glsdescwidth}|l|p{\glspagelistwidth}|}%
\end{supertabular}}%
}

```

8.16 Rollback v4.46 (glossary-superragged-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2020/03/19 v4.46 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{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{\glsgdescwidth}|%
>{\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{\glsgdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
}

\newglossarystyle{superragged3colheaderborder}{%
\setglossarystyle{superragged3colborder}%
\renewenvironment{theglossary}%
{
\tablehead{\hline
\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline}%
\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsgdescwidth}|%
>{\raggedright}p{\glspagelistwidth}|}%
{\end{supertabular}}%
}

\newglossarystyle{altsuperragged4col}{%
\renewenvironment{theglossary}%
{
\tablehead{}\tabletail{}%
\begin{supertabular}{|l>{\raggedright}p{\glsgdescwidth}l%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupeheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} &
\glossentrysymbol{##1} & ##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi

```

```

}
\newglossarystyle{altsuperragged4colheader}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsgdescwidth}l%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colborder}{%
  \setglossarystyle{altsuper4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {ll>{\raggedright}p{\glsgdescwidth}ll%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {ll>{\raggedright}p{\glsgdescwidth}ll%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}

```

8.17 Rollback v4.46 (glossary-tree-2020-03-19.sty)

Version 4.46 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2020/03/19 v4.46 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem

```



```

{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{%
\renewcommand*{\glossentry}[2]{%
\item\glssentryitem{##1}\glstreenamfmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamfmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space\glossentrysymbol{##2}}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*{\glossaryheader}{%

```

```

\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{tree}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}
\newglossarystyle{treegroup}{%
\setglossarystyle{tree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
\indexspace}%
}
\newglossarystyle{treehypergroup}{%
\setglossarystyle{tree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%

```

```

\renewenvironment{theglossary}%
  {\setlength{\parindent}{0pt}%
   \setlength{\parskip}{0pt plus 0.3pt}}%
  {}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \hangindent0pt\relax
  \parindent0pt\relax
  \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
  \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
  \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstarget{##2}{\strut}%
  \glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt
    {\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \gls@tmplen=0pt\relax
  \forallglossaries[#1]{\@gls@type}%
  {%
    \forglsentries[\@gls@type]{\@glo@label}%
    {%
      \ifglshasparent{\@glo@label}%
      {}%
    }%
  }%
}

```

```

\settowidth{\dimen@}{%
  {\glstreenamefmt{\glstentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
  \gls@tmplen=\dimen@
  \letcs{\@glswidestname}{glo@glstdetoklabel{\@glo@label}@name}%
\fi
}%
}%
}%
}
\newcommand*{\glssetwidest}[2][0]{%
  \expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
    #2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
  \makebox[#1][l]{#2}%
}
\newglossarystyle{alttree}{%
  \renewenvironment{theglossary}%
    {\def\@gls@prevlevel{-1}%
     \mbox{}\par}%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \ifnum\@gls@prevlevel=0\relax
    \else
      \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname}\space}%
    \fi
    \hangindent\glstreeindent
    \parindent\glstreeindent
    \makebox[0pt][r]{\glstreenamebox{\glstreeindent}{%
      \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
    \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
    \glossentrydesc{##1}\glspostdescription \space ##2\par
    \def\@gls@prevlevel{0}%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
      \@ifundefined{@glswidestname\romannumeral##1}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname}\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
      \ifnum\@gls@prevlevel<##1\relax
        \setlength{\glstreeindent}{\gls@tmplen

```

```

\addtolength\glstreeindent\parindent
\parindent\glstreeindent
\else
\@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\@glswidestname\space}}}%
\settowidth{\glstreeindent}{\glstreenamefmt{%
\csname @glswidestname\romannumeral\@gls@prevlevel
\endcsname\space}}}%
\addtolength\parindent{-\glstreeindent}%
\setlength\glstreeindent\parindent
\fi
\fi
\hangindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
\indexspace}}

```

8.18 Rollback v4.49 (glossaries-2021-11-01.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareCurrentRelease{v4.49}{2021-11-01}
\ProvidesPackage{glossaries}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\RequirePackage{textcase}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}%
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
{%
  \@gls@docloadedtrue
}%
}%
{%
  \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}{[\gls@debug@val\gls@debug@nr]{%
{true,false,showtargets,showaccsupp}[true]{%
  \ifcase\gls@debug@nr\relax
    % debug=true
    \@gls@debugtrue
    \renewcommand*{\GlossariesWarning}[1]{%
      \PackageWarning{glossaries}{##1}%
    }%
    \renewcommand*{\GlossariesWarningNoLine}[1]{%
      \PackageWarningNoLine{glossaries}{##1}%
    }%
    \let\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]{%
  \glsshowtargetsymbolsmarginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbols{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont{\ttfamily\footnotesize}
\newcommand*\@glsshowtarget}[1]{%
\newrobustcmd*\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%

```

```

\endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{}
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@glsshowtarget{/#2 (#3)}}%
  {\@glsshowtarget{/#2 (#3) [#1]}}%
}
\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinde}x{%
  [\gls@seenoinde@val\gls@seenoinde@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinde@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@glossarysec}{section}}%
  {\newcommand*{\@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
  subsection,subsubsection,paragraph,subparagraph}[section]{%

```



```

\renewcommand*{\@glossarysec}{#1}}
\newcommand*{\@glossarysecstar}{*}%
\newcommand*{\@glossaryseclabel}{}%
\newcommand*{\glsautoprefix}{%
\define@choicekey{glossaries.sty}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]{%
false,nolabel,autolabel,nameref}[nolabel]{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{%
\label{\glsautoprefix@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
\renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{%
\gls@savenumberlistfalse
\newcommand*\@glo@seeautonumberlist{}
\@gls@declareoption{seeautonumberlist}{%
\renewcommand*\@glo@seeautonumberlist{%
\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\gls@esclocationstrue
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}[=v4.49]}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}}

```

```

\IfFileExists{supertabular.sty}{%
  \newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}[=v4.49]}{%
    \newcommand*{\@gls@loadsuper}{}}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}}
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}[=v4.49]}
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
    \ifdefstring{\@glossary@default@style}{list}%
    {\let\@glossary@default@style\relax}%
    }%
  }%
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}[=v4.49]}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%
    \else
    \newcounter{glossaryentry}[\@gls@counterwithin]%
    \fi
    \def\theHglossaryentry{\currentglossary.\theglossaryentry}%
  }%
  {%
  }%
}
\fi
}

```

```

\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
}%
}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glssubentrycounterfalse
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\@gls@prestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \def\@gls@defsort##1##2{%
      \ifx\@glo@sort\@gls@defaultsort
        \let\@glo@sort\@glo@name
      \fi
      \let\glsdosanitizesort\@gls@sanitizesort
      \glsprestandardsort{\@glo@sort}{##1}{##2}%
      \expandafter\protected@xdef\csname glo###2@sort\endcsname{\@glo@sort}%
    }%
  \def\@gls@setsort##1{%
    \let\@glo@check@sortallowed\@gobble
  }
}
\@gls@setupsort@standard
\newcommand*\glssortnumberfmt[1]{%
  \ifnum#1<100000 0\fi

```

```

\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*{\@gls@setupsort@def}{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
{}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glssortnumberfmt
\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@use}{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi

```

```

}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}
\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
\ifglsacronym
\renewcommand{\@gls@do@acronymsdef}{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*{\acronymtype}{acronym}%
\newcommand*{\gls@tr@set@acronym@toctitle}{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
\else
\let\@gls@do@acronymsdef\relax
\fi
}
\AtBeginDocument{%
\ifglsacronym
\ifbool{glscompatible-3.07}%
{}%
{%
\providecommand*{\printacronyms}[1]{}%
}
}

```

```

        \printglossary[type=\acronymtype,#1]]%
    }%
\fi
}
\newcommand*{\@gls@do@acronymsdef}{%
\@gls@declareoption{acronyms}{%
\glsacronymtrue
\def\@gls@do@acronymsdef{%
\DeclareAcronymList{acronym}%
\newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
\renewcommand*{\acronymtype}{acronym}%
\newcommand*{\gls@tr@set@acronym@toctitle}{%
\translatelet{\glossarytoctitle}{Acronyms}%
}%
}%
}
\newcommand*{\@glsacronymlists}{%
\newcommand*{\@addtoacronymlists}[1]{%
\ifx\@glsacronymlists\@empty
\protected@xdef\@glsacronymlists{#1}%
\else
\protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
\fi
}
\newcommand*{\DeclareAcronymList}[1]{%
\glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand{\glsIfListOfAcronyms}[1]{%
\protected@edef\@do@gls@islistofacronyms{%
\noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
\@do@gls@islistofacronyms
}
\newcommand{\@gls@islistofacronyms}[4]{%
\def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
\def\@gls@before{##1}\def\@gls@after{##2}}%
\gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
\ifx\@gls@after\@nnil
#4%
\else
#3%
\fi
}
\newif\if@glsisacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
\glsIfListOfAcronyms{#1}%
{\@glsisacronymlisttrue}{\@glsisacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
\renewcommand*{\@glsacronymlists}{#1}%
}

```

```

\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\glscounter}{page}
\define@key{glossaries.sty}{counter}{%
  \renewcommand*{\glscounter}{#1}%
}
\newcommand*{\@gls@nohyperlist}{}
\newcommand*{\GlsDeclareNoHyperList}[1]{%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \appto\@gls@nohyperlist{,#1}%
  }%
}
\define@key{glossaries.sty}{nohypertypes}{%
  \GlsDeclareNoHyperList{#1}%
}
\newcommand*{\GlossariesWarning}[1]{%
  \PackageWarning{glossaries}{#1}%
}
\newcommand*{\GlossariesWarningNoLine}[1]{%
  \PackageWarningNoLine{glossaries}{#1}%
}
\newcommand{\glosortentrieswarning}{%
  \typeout{Using TeX to sort glossary entries---this may
  take a while}%
}
\@gls@declareoption{nowarn}{%
  \if@gls@debug
    \GlossariesWarning{Warnings can't be suppressed in debug mode}%
  \else
    \renewcommand*{\GlossariesWarning}[1]{}%
    \renewcommand*{\GlossariesWarningNoLine}[1]{}%
    \renewcommand*{\glosortentrieswarning}{}%
    \renewcommand*{\@gls@missinglang@warn}[2]{}%
  \fi
}
\newcommand*{\@gls@missinglang@warn}[2]{%
  \PackageWarningNoLine{glossaries}%
  {No language module detected for '#1'.\MessageBreak
  Language modules need to be installed separately.\MessageBreak
  Please check on CTAN for a bundle called\MessageBreak
  'glossaries-#2' or similar}%
}
\@gls@declareoption{nolangwarn}{%
  \renewcommand*{\@gls@missinglang@warn}[2]{}%
}

```

```

\newcommand*{\@gls@warnonglossdefined}{%
  \GlossariesWarning{Overriding \string\printglossary}%
}
\newcommand*{\@gls@warnontheGLOSSdefined}{%
  \GlossariesWarning{Overriding 'theGLOSS' environment}%
}
\@gls@declareoption{noredefwarn}{%
  \renewcommand*{\@gls@warnonglossdefined}{}%
  \renewcommand*{\@gls@warnontheGLOSSdefined}{}%
}
\newcommand*{\@gls@sanitizedesc}{%
}
\newcommand*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifgls@sanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
  {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
}

```



```

\@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
\ifdefined\@glo@sort
{}%
{%
\expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
}%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
\bgroup
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egroup
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent\@secondoftwo
\let\@text@composite\@secondoftwo
\let\@tabacckludge\@secondoftwo
\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let\'\'@firstofone
\let\'\'@firstofone
\let\~\@firstofone
\let\"\"@firstofone
\let\u\@firstofone
\let\t\@firstofone
\let\d\@firstofone
\let\r\@firstofone
\let\=\@firstofone
\let\.\@firstofone
\let\~\@firstofone
\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%

```

```

\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[glS]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifglS@sanitize@symbol
\glssetnoexpandfield{symbol}%
\glssetnoexpandfield{symbolplural}%
\else
\glssetexpandfield{symbol}%
\glssetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
\ifglssanitizesort
\glssetnoexpandfield{sortvalue}%
\renewcommand*{\@glS@noidx@setsanitizesort}{%
\glssanitizesorttrue
\glssetnoexpandfield{sortvalue}%
}%
\else
\glssetexpandfield{sortvalue}%

```

```

\renewcommand*\@gls@noidx@setsanitizesort}{%
  \glssanitizesortfalse
  \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}%
      \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
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
  {\protected@eappto\@glsl@fulllist{\@ignored@glossaries}}%
  \forallglseentries[\@glsl@fulllist]{\@glseentry}%
  {%
    \ifdefempty{\@glseentry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glsl@thisentryname{%
      \else
        \letcs{\@glsl@thisentryname}{glo\@glsdetoklabel{\@glseentry}@name}%
        \@onelevel@sanitize\@glsl@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@glseentry\do@write@glslabelsname{\@glsl@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*{\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^^I12
  \gdef\@write@glslabelsname#1{\^^I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifglslautomake

```

```

\newcommand{\gls@automake@nr}{1}
\define@choicekey{glossaries.sty}{automake}%
  [\gls@automake@val\gls@automake@nr]{true,false,immediate}[true]{%
  \ifnum\gls@automake@nr=1\relax
    \glsautomakefalse
  \else
    \glsautomaketrue
  \fi
  \ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
      \PackageError{glossaries}{You must use
        \string\makeglossaries\space with automake=true}
        {%
          Either remove the automake=true setting or
          add \string\makeglossaries\space to your document preamble.%
        }%
    }%
  \else
    \renewcommand*{\@gls@doautomake}{}%
  \fi
}
\glsautomakefalse
\newcommand*{\@gls@doautomake}{%}
\AtEndDocument{\@gls@doautomake}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifgls savewrites
    \renewcommand*{\glswritefiles}{\@glswritefiles}%
  \else
    \let\glswritefiles\@empty
  \fi
}
\gls savewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%}
\ifbool{glscompatible-2.07}%
  {%
    \booltrue{glscompatible-3.07}%
  }%
  {%}
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\def\makeglossary{%
  \GlossariesWarning{Use of \string\makeglossary\space with
    glossaries.sty is \MessageBreak deprecated. Use \string\makeglossaries\space
    instead. If you \MessageBreak need the original definition of
    \string\makeglossary\space use \MessageBreak the package options

```



```

kernelglossredefs=false (to \MessageBreak restore the former definition of
\string\makeglossary) and \MessageBreak nomain (if the file extensions cause a
conflict))}%
\makeglossaries
}
\newcommand*{\@gls@override@glossary}[1][main]{%
\GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict))}%
\gls@glossary{#1}%
}
\if@gls@docloaded
\else
\def\glossary{\@gls@override@glossary}
\fi
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@debug@val\gls@debug@nr]{true,false,nowarn}[true]%
{%
\ifcase\gls@debug@nr\relax
\def\glossary{\@gls@override@glossary}%
\def\makeglossary{%
\GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict))}%
\makeglossaries
}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{}
\newcommand*{\@gls@symbolsdef}{%

```

```

\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]]{%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
\newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
\newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
\newcommand*{\gls@tr@set@numbers@toctitle}{%
\translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
}%
}%
\@gls@declareoption{index}{%
\ifx\@gls@do@indexdef\@empty
\let\@gls@do@indexdef\@gls@indexdef
\fi
}
\@gls@declareoption{noglossaryindex}{%
\let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
\newglossary[ilg]{index}{ind}{idx}{\indexname}%
\newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
\newcommand*{\newterm}[2][\%
\newglossaryentry{##2}%
{type={index},name={##2},description={\nopostdesc},##1}}
\let\@gls@do@indexdef\relax
}%
}%
\@for\CurrentOption :=\@declaredoptions\do{%
\ifx\CurrentOption\@empty
\else
\@expandtwoargs
\in@ {\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
\ifin@
\@use@option
\expandafter \let\csname ds@\CurrentOption\endcsname\@empty
\fi
\fi
}
\ProcessOptionsX
\RequirePackage{glossaries-compatible-307}[v4.49]
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,nolong,nosuper,notree,nostyles,%

```

```

    nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsentrycounter
    \let\@gls@doentrycounterdef\relax
  \else
    \let\@gls@doentrycounterdef\@gls@define@glossaryentrycounter
  \fi
  \ifglssubentrycounter
    \let\@gls@dosubentrycounterdef\relax
  \else
    \let\@gls@dosubentrycounterdef\@gls@define@glossarysubentrycounter
  \fi
  \setkeys{glossaries.sty}{#1}%
  \@gls@setacrstyle
  \@gls@setupshortcuts
  \@gls@do@acronymsdef
  \@gls@do@numbersdef
  \@gls@do@symbolssdef
  \@gls@do@indexdef
  \@gls@doentrycounterdef
  \@gls@dosubentrycounterdef
}
\ifthenelse{\equal{\glscounter}{section}}{%
  {%
    \ifcsundef{chapter}{}%
    {%
      \let\@gls@old@chapter\@chapter
      \def\@chapter[#1]#2{\@gls@old@chapter[#1]}{#2}%
      \ifcsundef{hyperdef}{\hyperdef{section}{\thesection}}}%
    }%
  }%
  {}
\newcommand*{\@gls@onlypremakeg}{%
\newcommand*{\@onlypremakeg}[1]{%
  \ifx\@gls@onlypremakeg\@empty

```

```

\def\@gls@onlypremakeg{#1}%
\else
\expandafter\toks@\expandafter{\@gls@onlypremakeg}%
\edef\@gls@onlypremakeg{\the\toks@,\noexpand#1}%
\fi
}
\newcommand*{\@disable@onlypremakeg}{%
\@for\@thiscs:=\@gls@onlypremakeg\do{%
\expandafter\@disable@premakecs\@thiscs%
}}
\newcommand*{\@disable@premakecs}[1]{%
\def#1{\PackageError{glossaries}{\string#1\space may only be
used before \string\makeglossaries}{You can't use
\string#1\space after \string\makeglossaries}}%
}
\providecommand*{\glossaryname}{Glossary}
\providecommand*{\acronymname}{Acronyms}
\newcommand*{\glssettoctitle}[1]{%
\def\glossarytoctitle{\csname @glotype@#1@title\endcsname}}
\providecommand*{\entryname}{Notation}
\providecommand*{\descriptionname}{Description}
\providecommand*{\symbolname}{Symbol}
\providecommand*{\pagelistname}{Page List}
\providecommand*{\glsymbolsgroupname}{Symbols}
\providecommand*{\glsnumbersgroupname}{Numbers}
\newcommand*{\glspluralsuffix}{s}
\newcommand*{\glsacrpluralsuffix}{\glspluralsuffix}
\newcommand*{\glsupacrpluralsuffix}{\glsup{t}\glsacrpluralsuffix}
\providecommand*{\seename}{see}
\providecommand*{\andname}{\&}
\newcommand*{\RequireGlossariesLang}[1]{%
\ifundefined{ver@glossaries-#1.ldf}{\input{glossaries-#1.ldf}}{}}%
}
\newcommand*{\ProvidesGlossariesLang}[1]{%
\ProvidesFile{glossaries-#1.ldf}%
}
\newcommand*{\addglossarytocaptions}[1]{}
\ifglstranslate
\RequirePackage{tracklang}
\@gls@usetranslator
\ifpackageloaded{translator}
{%
\ifboolexpr
{
test {\ifdefstring{\trans@languages}{English}}
and not
test {\ifdefstring{bbl@loaded}{english}}
}
{%
\let\glsifusetranslator\@secondoftwo

```

```

}%
{%
  \usedictionary{glossaries-dictionary}%
  \renewcommand*{\addglossarytocaptions}[1]{%
    \ifcsundef{captions#1}{}%
    {%
      \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
      \expandafter\toks@\expandafter{\@gls@tmp
        \renewcommand*{\glossaryname}{\translate{Glossary}}}%
    }%
    \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
  }%
}%
}%
{}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%

```

```

        \renewcommand*{\glssymbolsgroupname}{%
          \translate{Symbols (glossaries)}}%
        \renewcommand*{\glsnumbersgroupname}{%
          \translate{Numbers (glossaries)}}%
      }{}%
\fi
\DeclareRobustCommand*{\nopostdesc}{}
\newcommand*{\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
  \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]{\glsnumber{#1}}%
}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaulttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls@glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls@glossarymark}[1]{\glossarymark{#1}}
}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUChead{#1}}{\memUChead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls@glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
      \else

```

```

        \@mkboth{#1}{#1}%
    \fi
}
}
\providecommand{\glossarymark}[1]{%
    \ifglsucmark
        \@mkboth{\mfirstucMakeUppercase{#1}}{\mfirstucMakeUppercase{#1}}%
    \else
        \@mkboth{#1}{#1}%
    \fi
}
\newcommand*{\setglossarysection}[1]{%
\setkeys{glossaries.sty}{section=#1}}
\newcommand*{\@glossarysection}[2]{%
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname[#1]{#2}%
    }%
    {%
        \csname\@glossarysec\endcsname*{#2}%
        \@gls@toc{#1}{\@glossarysec}%
    }%
    \@glossaryseclabel
}
\newcommand*{\@p@glossarysection}[2]{%
    \glsclearpage
    \phantomsection
    \ifdefempty\@glossarysecstar
    {%
        \csname\@glossarysec\endcsname{#2}%
    }%
    {%
        \@gls@toc{#1}{\@glossarysec}%
        \csname\@glossarysec\endcsname*{#2}%
    }%
    \@glossaryseclabel
}
\newcommand*{\gls@docclearpage}{%
    \ifthenelse{\equal{\@glossarysec}{chapter}}{%
    {%
        \ifcsundef{cleardoublepage}%
        {%
            \clearpage
        }%
        {%
            \ifcsdef{if@openright}%
            {%
                \if@openright
                \cleardoublepage
            }%
        }%
    }%
    }%
}

```



```

\else
\clearpage
\fi
}%
{%
\cleardoublepage
}%
}%
}%
{%
}%
}
\newcommand*{\glsclearpage}{\gls@docclearpage}
\newcommand*{\@gls@toc}[2]{%
\ifglstoc
\ifglsnumberline
\addcontentsline{toc}{#2}{\protect\numberline{#1}}%
\else
\addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*{\glsnoindywarning}[1]{%
\GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*{\glsnomakeindexwarning}[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\edef\xdyattributes{\string"default\string"}%
\fi
\ifglsxindy
\edef\xdyattributelist{%
\fi
\ifglsxindy
\def\xdylocref{}
\fi
\newcommand*{\@gls@ifinlist}[4]{%
\def\@do@ifinlist##1,#1,##2\end@doifinlist{%
\def\@gls@listsuffix{##2}%
\ifx\@gls@listsuffix\@empty
#4%
\else
#3%
\fi
}%
\@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
\newcommand*{\@xdycounters}{\glscounter}
\newcommand*{\GlsAddXdyCounters}[1]{%

```

```

\@for\@gls@ctr:=#1\do{%
  \edef\@do@addcounter{%
    \noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
    {%
      \noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
        \noexpand\@gls@ctr}%
    }%
  }%
  \do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
  \glsnoxindywarning\GlsAddXdyAttribute
}
\fi
\newcommand*\@disabled@glsaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}
\ifglsxindy
\newcommand*\@glsaddxdyattribute[2]{%
  \edef\@xdyattributes{\@xdyattributes ^^J \string"#1\string" ^^J
    \string"#2#1\string"%
  \expandafter\toks@\expandafter{\@xdylocref}%
  \edef\@xdylocref{\the\toks@ ^^J%
    (markup-locref
  :open \string"\glstildechar n%
    \expandafter\string\csname glsX#2X#1\endcsname
    \string" ^^J
  :close \string"\string" ^^J
  :attr \string"#2#1\string")}%
  \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
    \setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
  }%
}
\newcommand*\GlsAddXdyAttribute[1]{%
  \ifx\@xdyattributelist\@empty
    \edef\@xdyattributelist{#1}%
  \else
    \edef\@xdyattributelist{\@xdyattributelist,#1}%
  \fi
  \@for\@this@counter:=\@xdycounters\do{%
    \protected@edef\gls@do@addxdyattribute{%
      \noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
    }
    \gls@do@addxdyattribute

```

```

    }%
    \let\GlsAddXdyCounters\@disabled@gl saddx dy counters
  }
  \@onlypremakeg\GlsAddXdyAttribute
\else
  \newcommand*\GlsAddXdyAttribute[1]{%
    \gl snoxindywarning\GlsAddXdyAttribute}
\fi
\ifgl sxindy
\newcommand*{\@gl s@addpredefinedattributes}{%
  \GlsAddXdyAttribute{gl snumberformat}
  \GlsAddXdyAttribute{textrm}
  \GlsAddXdyAttribute{textsf}
  \GlsAddXdyAttribute{texttt}
  \GlsAddXdyAttribute{textbf}
  \GlsAddXdyAttribute{textmd}
  \GlsAddXdyAttribute{textit}
  \GlsAddXdyAttribute{textup}
  \GlsAddXdyAttribute{textsl}
  \GlsAddXdyAttribute{textsc}
  \GlsAddXdyAttribute{emph}
  \GlsAddXdyAttribute{gl shypernumber}
  \GlsAddXdyAttribute{hyper rm}
  \GlsAddXdyAttribute{hypersf}
  \GlsAddXdyAttribute{hypertt}
  \GlsAddXdyAttribute{hyperbf}
  \GlsAddXdyAttribute{hypermd}
  \GlsAddXdyAttribute{hyperit}
  \GlsAddXdyAttribute{hyperup}
  \GlsAddXdyAttribute{hypersl}
  \GlsAddXdyAttribute{hypersc}
  \GlsAddXdyAttribute{hyperemph}
  \GlsAddXdyAttribute{gl signore}
}
\else
  \let\@gl s@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{}
\ifgl sxindy
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \edef\@xdyuseralphabets{%
      \@xdyuseralphabets ~^J
      (define-alphabet "#1" (#2))}}
\else
  \newcommand*{\GlsAddXdyAlphabet}[2]{%
    \gl snoxindywarning\GlsAddXdyAlphabet}
\fi
\ifgl sxindy
  \newcommand*{\@gl s@xdy@locationlist}{%
    roman-page-numbers,%

```

```

    Roman-page-numbers,%
    arabic-page-numbers,%
    alpha-page-numbers,%
    Alpha-page-numbers,%
    Appendix-page-numbers,%
    arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@gls@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}}%
\@onelevel@sanitize\@gls@tmp
\ifx\@gls@tmp\@gls@roman
    \expandafter
        \edef\csname @gls@xdy\Lclass@roman-page-numbers\endcsname{%
            \string"roman-numbers-lowercase\string"%
        }%
\else
    \expandafter
        \edef\csname @gls@xdy\Lclass@roman-page-numbers\endcsname{%
            :sep \string"\@gls@roman\string"%
        }%
\fi
\expandafter\def\csname @gls@xdy\Lclass@Roman-page-numbers\endcsname{%
    \string"roman-numbers-uppercase\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@arabic-page-numbers\endcsname{%
    \string"arabic-numbers\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@alpha-page-numbers\endcsname{%
    \string"alpha\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@Alpha-page-numbers\endcsname{%
    \string"ALPHA\string"%
}%
\expandafter\def\csname @gls@xdy\Lclass@Appendix-page-numbers\endcsname{%
    \string"ALPHA\string"
    :sep \string"\@gls@alphacompositor\string"
    \string"arabic-numbers\string"%
}
\expandafter\def\csname @gls@xdy\Lclass@arabic-section-numbers\endcsname{%
    \string"arabic-numbers\string"
    :sep \string"\glscompositor\string"
    \string"arabic-numbers\string"%
}%
\def\@xdyuserlocationdefs{}
\def\@xdyuserlocationnames{}
\fi
\ifglxsindy

```

```

\newcommand\@xdycrossrefhook{}
\fi
\ifglxsindy
\newcommand*\GlsAddXdyLocation}[3][{}]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\glssopenbrace\string" #3
:sep \string"\glsclosebrace\string"))
}%
\else
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep "\glssopenbrace"
#1
:sep "\glsclosebrace\glssopenbrace" #3
:sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
\@xdyuserlocationnames^^J\space\space\space
\string"#2\string"}%
}
\@onlypremakeg\GlsAddXdyLocation
\else
\newcommand*\GlsAddXdyLocation}[2]{%
\glsnxindywarning\GlsAddXdyLocation}
\fi
\ifglxsindy
\def\@xdylocationclassorder{^^J\space\space\space
\string"roman-page-numbers\string"^^J\space\space\space
\string"arabic-page-numbers\string"^^J\space\space\space
\string"arabic-section-numbers\string"^^J\space\space\space
\string"alpha-page-numbers\string"^^J\space\space\space
\string"Roman-page-numbers\string"^^J\space\space\space
\string"Alpha-page-numbers\string"^^J\space\space\space
\string"Appendix-page-numbers\string"
\@xdyuserlocationnames^^J\space\space\space
\string"see\string"
}
\fi
\ifglxsindy
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\def\@xdylocationclassorder{#1}}
\else
\newcommand*\GlsSetXdyLocationClassOrder[1]{%
\glsnxindywarning\GlsSetXdyLocationClassOrder}

```

```

\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))}%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnxindywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnxindywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnxindywarning\GlsSetXdyStyles}
\fi
\newcommand*\findrootlanguage{-}
\def\@xdylanguage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnxindywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }

```

```

\AtBeginDocument{%
  \ifx\gls@codepage\empty
    \ifpackageloaded{fontspec}{\def\gls@codepage{utf8}}{}%
  \fi
}
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnoxywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglxindy@glxnumbers
    \def\@xdylettergroups{(define-letter-group
      \string\glxnumbers\string^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"
        \string"2\string" \string"3\string" \string"4\string"
        \string"5\string" \string"6\string" \string"7\string"
        \string"8\string" \string"9\string")^^J\space\space\space
      \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
  \expandafter\toks@\expandafter{\@xdylettergroups}%
  \protected@edef\@xdylettergroups{\the\toks@^^J%
    (define-letter-group \string"#1\string"^^J\space\space\space#2)}}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
  \@for#2:=#1\do{\ifx#2\empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
  \@for#1:=\@glsacronymlists\do{\ifx#1\empty\else#2\fi}%
}
\newcommand*\forglentries[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glo@list@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{\fi}%
  }%
}
\newcommand*\forallglentries[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forglentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand*\ifglossaryexists{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}

```

```

\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{@glotype@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
  }
}

```



```

\protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
\expandafter\forlentries\expandafter
  [\csname glo@\@gls@thislabel @type\endcsname]
{\glo@label}%
{%
  \letcs\glo@parent{glo@\glo@label @parent}%
  \ifdefequal\@gls@thislabel\glo@parent
  {%
    \def\do@glshaschildren{#2}%
    \@endfortrue
  }%
  {}%
}%
\do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcseempty{glo@\glsdetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcseempty{glo@\glsdetoklabel{#1}@desc}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasdescsuppressed}[3]{%
  \ifcsequal{glo@\glsdetoklabel{#1}@desc}{\no@post@desc}%
  {#2}%
  {#3}%
}
\newrobustcmd*{\ifglshassymbol}[3]{%
  \letcs{\@glo@symbol}{glo@\glsdetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
  {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
    {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@\glsdetoklabel{#1}@long}%
  \ifdefempty\@glo@long
  {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
    {#3}%
    {#2}%
  }%
}

```

```

    }%
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo\glsdetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
      {#3}%
    {#2}%
  }%
}
}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo\glsdetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        {%
          \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
              #4%
            }%
          {%
            \let\glscurrentfieldvalue\@glo@thisvalue
            #3%
          }%
        }%
      }%
    }%
    {%
      \@gls@fetchfield{\@gls@thisfield}{#1}%
      \ifdef\@gls@thisfield
        {%
          \letcs{\@glo@thisvalue}{glo\glsdetoklabel{#2}@#1}%
          \ifdefempty\@glo@thisvalue
            {%
              #4%
            }%
          {%
            \ifdefequal\@glo@thisvalue\@gls@default@value
              {%
                #4%
              }%
            {%
              \let\glscurrentfieldvalue\@glo@thisvalue
              #3%
            }%
          }%
        }%
      }%
    }%
  }%

```

```

    }%
  }%
} %
{ %
  \GlossariesWarning{Unknown entry field ‘#1’}%
  #4%
} %
} %
} %
}
\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{@glo@types}{{,}}
\newcommand*{\@gls@provide@newglossary{%
  \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
  \let\@gls@provide@newglossary\relax
}
\newcommand*{\defglsentryfmt}[2][\glsdefaultttype]{%
  \csgdef{gls@#1@entryfmt}{#2}%
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{@gls@forbidtexext}[1]{%
  \ifboolexpr{test {\ifdefstring{#1}{tex}}
    or test {\ifdefstring{#1}{TEX}}}
  {%
    \def#1{nottex}%
    \PackageError{glossaries}%
      {Forbidden ‘.tex’ extension replaced with ‘.nottex’}%
      {I’m sorry, I can’t allow you to do something so reckless.\MessageBreak
        Don’t use ‘.tex’ as an extension for a temporary file.}%
  }%
  {%
  }%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{}
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaultttype
    {%
      \gdef\glsdefaultttype{#2}%
    }{%
    }%
    \toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
    \expandafter\gdef\csname glolist@#2\endcsname{,}%
    \expandafter\edef\csname @gloptype@#2@log\endcsname{#1}%
    \expandafter\edef\csname @gloptype@#2@in\endcsname{#3}%
  }

```

```

\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtextext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtextext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
\defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}{%
{\@gls@setcounter{#2}[\glscounter]}}%
}%
{%
\gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
\newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
\expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
\ifglxsindy
\GlsAddXdyCounters{#2}%
\fi
}
\newcommand*{\@gls@getcounter}[1]{%
\csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
\ifdefempty\@ignored@glossaries
{%
\protected@edef\@ignored@glossaries{#1}%
}%
{%
\protected@eappto\@ignored@glossaries{,#1}%
}%
\csgdef{glolist@#1}{,}%
\ifcsundef{gls@#1@entryfmt}%

```

```

{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}%
{}%
\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*{\@gls@nohyperlist}{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*{\@ignored@glossaries}{}
\newcommand*{\ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
    {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*{\@gls@default@value}{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}

```

```

\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
\def\@glo@counter{#1}%
}%
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}%
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}%
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}%
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {‘\gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}%
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
  [\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}%
\newcommand*{\@gls@savenonumberlist}[1]{%
\newcommand*{\@gls@initnonumberlist}{%
\newcommand*{\@gls@storenonumberlist}[1]{%

```

```

\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
      {%
        \cslet\glo@\glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
      }%
      {}%
    }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@name}{%
  \PackageError{glossaries}{name key required in
    \string\newglossaryentry\space for entry '\@glo@label'}{You
    haven't specified the entry name}}

```

```

\newcommand*\@glsnodedesc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*\@glsdefaultplural{}
\newcommand*\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'.
      Maybe makeglossaries + rerun required}%
  \else
    \PackageError{glossaries}%
    {Package option 'savenumberlist=true' required}%
    {%
      You must use the 'savenumberlist' package option
      to reference location lists.%
    }%
  \fi
}
\newcommand*\@glsdefaultsort{}\@glo@name
\newcount\gls@level
\newcommand*\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand*\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}\@gls@value}%
    }%
    {%
  }

```



```

        \@gls@noexpand@field{#2}{#3}{#4}%
    }%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \newcommand*{\gls@defdocnewglossaryentry}{%
    \let\newglossaryentry\new@glossaryentry
  }
  \newrobustcmd{\provideglossaryentry}[2]{%
    \ifglsentryexists{#1}%
    {%
      {%
        \gls@defglossaryentry{#1}{#2}%
      }%
    }
  }
  \@onlypreamble{\provideglossaryentry}
  \newcommand*{\gls@glsdefs@labellist}{}
  \newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
  \newcommand*{\gls@docdef@labellist}{}
  \newcommand*{\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
  \newrobustcmd{\new@glossaryentry}[2]{%
    \ifundef\@gls@deffile
    {%
      \global\newwrite\@gls@deffile
      \immediate\openout\@gls@deffile=\jobname.glsdefs
    }%
    {%
      \ifglsentryexists{#1}%
      {%
        \xifinlist{#1}{\gls@docdef@labellist}%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in the
            document environment (ignoring new definition)}%
        }%
        {%
          \xifinlist{#1}{\gls@glsdefs@labellist}%
          {%
            \gls@defglossaryentry{#1}{#2}%
            \gls@docdef@addlabel{#1}%
            \@gls@writedef{#1}%
          }%
          {%
            \GlossariesWarning{Entry ‘#1’ has already been defined in
              the preamble (in which case remove the duplicate definition)
              or you have just updated to glossaries.sty
              v4.47 (in which case delete ‘\jobname.glsdefs’)}%
          }%
        }%
      }%
    }%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }
  }

```

```

\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*\gls@begindocdefs{%
\@gls@enablesavenonnumberlist
\edef\@gls@restoreat{\noexpand\catcode'\noexpand\@=\number\catcode'\@}\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{\gls@restoreat}{\gls@restoreat}
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}
\newcommand*\gls@defdocnewglossaryentry[2]{%
\ifglsentryexists{#1}%
{\GlossariesWarning{entry '#1' has already been defined in the preamble
(if you have moved the definition there from the document
environment, delete \jobname.glsdefs)}}%
{%
\gls@docdef@addlabel{#1}%
\gls@defdocnewglossaryentry{#1}{#2}%
}%
}
\newcommand*\@gls@writedef[1]{%
\immediate\write\@gls@deffile
{%
\string\gls@defdocnewglossaryentry{\gls@detoklabel{#1}}\glspercentchar^^J%
\expandafter\@gobble\string\{\glspercentchar%
}%
\@for\@gls@map:=\@gls@keymap\do
{%
\letcs\gls@value{gls@detoklabel{#1}}\expandafter\@secondoftwo\@gls@map}%
\ifdef\gls@value
{%
\@onelevel@sanitize\gls@value
\immediate\write\@gls@deffile
{%
\expandafter\@firstoftwo\@gls@map
=\expandafter\@gobble\string\{\gls@value\expandafter\@gobble\string\},%
\glspercentchar
}%
}%
}%
}
\gls@writedefhook
\immediate\write\@gls@deffile
{%
\glspercentchar^^J%

```

```

\expandafter\@gobble\string\}\glpercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
}

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
}%

```

```

\expandafter\newcommand\expandafter*\expandafter
{\csname @gls@user@#1\endcsname}[2] [] {%
  \new@ifnextchar [%
    {\csuse{@gls@user@#1@}{##1}{##2}}}%
    {\csuse{@gls@user@#1@}{##1}{##2} []}}%
\csdef{@gls@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
  \expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#6' as helper command
   '\expandafter\string\csname @Gls@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @Gls@user@#1\endcsname}[2] [] {%
  \new@ifnextchar [%
    {\csuse{@Gls@user@#1@}{##1}{##2}}}%
    {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
\csdef{@Gls@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @GLS@user@#1\endcsname}[2] [] {%
  \new@ifnextchar [%
    {\csuse{@GLS@user@#1@}{##1}{##2}}}%
    {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
\csdef{@GLS@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{\mfirstucMakeUppercase{#3{##2}##3}}%
}%
\newrobustcmd*{#7}{%
  \expandafter\@gls@hyp@opt\csname @GLS@user@#1\endcsname}%
}%
}%

```

```

    {%
      \PackageError{glossaries}{Key ‘#1’ already exists}{}%
    }%
  }
\newcommand{\glsfieldxddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csxdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfielddedef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \protected@csedef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
  }%
}

```

```

}%
{%
    \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand{\glsfieldfetch}[3]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \letcs#3{glo@\@glo@label @#2}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
    \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
    \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
    \fi
}
\newcommand{\ifglsfieldeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\ifglsfielddefeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \expandafter\ifdefstrequal
            \csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}

```



```

}
\newcommand{\ifglsfieldcseq}[5]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
}%
{%
\PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
}%
}%
}
\newcommand*{\glswritedefhook}{%
\newcommand*{\gls@assign@desc}[1]{%
\gls@assign@field{#1}{desc}{\@glo@desc}%
\gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
\glsdoifnoexists{#1}%
{%
\bggroup
\let\@org@newglossaryentryprehook\@newglossaryentryprehook
\long\def\@newglossaryentryprehook{%
\long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
\@org@newglossaryentryprehook
}%
\renewcommand*{\gls@assign@desc}[1]{%
\global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
\global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
}
\gls@defglossaryentry{#1}{#2}%
\egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
\ifglsentryexists{#1}{%
{\longnewglossaryentry{#1}{#2}{#3}}%
}
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
\let\GlsSetQuote\gls@nosetquote
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\glslabel\@glo@label
\let\@glo@name\@gls@name
\let\@glo@desc\@gls@desc
\let\@glo@descplural\@gls@descplural
\let\@glo@type\@gls@type

```

```

\let\@glo@symbol\@gls@default@value
\let\@glo@symbolplural\@gls@default@value
\let\@glo@text\@gls@default@value
\let\@glo@plural\@gls@default@value
\let\@glo@first\@gls@default@value
\let\@glo@firstplural\@gls@default@value
\let\@glo@sort\@gls@default@value
\let\@glo@counter\@gls@default@value
\def\@glo@see{}%
\def\@glo@parent{}%
\def\@glo@prefix{}%
\@gls@initnonumberlist
\def\@glo@useri{}%
\def\@glo@userii{}%
\def\@glo@useriii{}%
\def\@glo@useriv{}%
\def\@glo@userv{}%
\def\@glo@uservi{}%
\def\@glo@short{}%
\def\@glo@shortpl{}%
\def\@glo@long{}%
\def\@glo@longpl{}%
\@newglossaryentryprehook
\setkeys{glossentry}{#2}%
\ifundef\glsdefaulttype
{%
  \PackageError{glossaries}%
  {No default glossary type (have you used ‘nomain’ by mistake?)}%
  {If you use package option ‘nomain’ you must define
   a new glossary before you can define entries}%
}%
{}%
\gls@assign@field{\glsdefaulttype}{\@glo@label}{type}{\@glo@type}%
\protected@edef\@glo@type{\glsentrytype{\@glo@label}}%
\ifcsundef{glo@list@\@glo@type}%
{%
  \PackageError{glossaries}%
  {Glossary type ‘\@glo@type’ has not been defined}%
  {You need to define a new glossary type, before making entries
   in it}%
}%
{}%
\ifignoredglossary\@glo@type
{%
  \ifx\@glo@desc\@gls@nodesc
    \let\@glo@desc\@empty
  \fi
}%
{}%
}%

```

```

\protected@edef\@glo@list@{\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \@glo@list@{\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@glsnoname
        \expandafter\let\expandafter\@glo@name
        \csname glo@\@glo@parent @name\endcsname
      \ifx\@glo@plural\@gls@default@value
        \expandafter\let\expandafter\@glo@plural
        \csname glo@\@glo@parent @plural\endcsname
      \fi
    \fi
  }%
  {%
    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
    }%
    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%

```

```

        {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @text\endcsname}%
    {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @plural\endcsname}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
    \expandafter\gls@assign@field\expandafter
        {\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
        {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
{%
    \def\@glo@defaultcounter{\glscounter}%
}%
{%
    \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@glsnoname
    \@glsnoname
    \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\@glo@label @numberlist}%
{%
    \csxdef{glo@\@glo@label @numberlist}{%
        \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
{%
    \gls@storenonumberlist{\@glo@label}%
\def\@glo@desc{\@glo@first}%
\ifx\@glo@desc\@glo@desc
    \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@glsnodesc
    \@glsnodesc
    \let\@glo@desc\@gls@default@value

```

```

\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@@symbol
  \let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
  {\csname glo@\@glo@label @symbol\endcsname}
  {\@glo@label}{symbolplural}{\@glo@symbolplural}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
  \noexpand\let\expandafter\noexpand
    \csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
  \noexpand\let\expandafter\noexpand
    \csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
  \csdef{glo@\@glo@label @index}{}%
}
{%
  \do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
\ifdefvoid\@glo@see{%
{%
  \protected@edef\@do@glssee{%
    \noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
    \noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
  \@do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{%
\newcommand*{\@newglossaryentryprehook}{%
\newcommand*{\@newglossaryentryposthook}{%
\newcommand*{\@newglossaryentry@defcounters}{%
\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%

```

```

\protected@edef\glo@type{\csname glo@\glo@thislabel @type\endcsname}%
\def\glo@list{,}%
\forlssentries[\glo@type]{\glo@label}%
{%
  \ifdefequal\glo@thislabel\glo@label
    {\protected@eappto\glo@list{\glo@label,}}%
  }%
\cslet\glo@list\glo@type{\glo@list}%
\csdef\glo@\glo@thislabel @type}{#2}%
}
\ifglxindy
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\else
  \newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglxindy
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\else
  \newcommand*{\@glossarysubentryfield}{%
    \string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
  \protected@edef\glo@esclabel{#1}%
  \gls@checkmkidxchars\glo@esclabel
  \protected@edef\glo@sort{\csname glo@#1@sort\endcsname}%
  \gls@checkmkidxchars\glo@sort
  \gls@checkmkidxchars\glo@prefix
  \protected@edef\glo@parent{\csname glo@#1@parent\endcsname}%
  \ifglxindy
    \ifx\glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\glo@sort\string" %
        \string"\glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\glo@parent @index\endcsname
        (\string"\glo@sort\string" %
        \string"\glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\glo@parent\@empty
      \@onelevel@sanitize\glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \glo@sort\gls@actualchar\glo@prefix
        \@glossaryentryfield{\@glo@esclabel}%
      }%
    \fi
  \fi

```

```

\else
\expandafter\protected\def\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\AtBeginDocument{%
\ifpackageloaded{amsmath}%
{\let\gls@ifnotmeasuring\@gls@ifnotmeasuring}%
{}}%
}
\newcommand*{\@gls@ifnotmeasuring}[1]{%
\ifmeasuring@
\else
#1%
\fi
}
\newcommand*\gls@ifnotmeasuring[1]{#1}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\let\glsunset\@gobble#2}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
{}}%
}
\newcommand*{\glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*{\glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}%

```

```

}
\newcommand*{\glsunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glsunset{#1}%
    }%
  }%
}
\newcommand*{\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*{\@glslocalunset}{\@glslocalunset}
\newcommand*{\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[1]{\@glsentry}%
  {%

```



```

        \glsunset{\@glsentry}%
    }%
}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
    \forallglsentries[#1]{\@glsentry}%
    {%
        \glslocalunset{\@glsentry}%
    }%
}
\newcommand*{\@newglossaryentry@defcounters}{%
    \csdef{glo@\@glo@label @currcount}{0}%
    \csdef{glo@\@glo@label @prevcount}{0}%
}
\newcommand*{\glsenableentrycount}{%
    \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
    \renewcommand*{\gls@defdocnewglossaryentry}{%
        \renewcommand*{\newglossaryentry}[2]{%
            \PackageError{glossaries}{\string\newglossaryentry\space
                may only be used in the preamble when entry counting has
                been activated}{If you use \string\glsenableentrycount\space
                you must place all entry definitions in the preamble not in
                the document environment}%
        }%
    }%
}
\newcommand*{\glsentrycurrcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
    {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
    \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
    {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
    \@glsunset{##1}%
    \@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
    \@glslocalunset{##1}%
    \@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
    \@glsreset{##1}%
    \csgdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\renewcommand*{\@glslocalreset}[1]{%
    \@glslocalreset{##1}%
    \csdef{glo@\glsdetoklabel{##1}@currcount}{0}%
}%
\def\@cgls@##1##2[##3]{%
    \ifnum\glsentryprevcount{##2}=1\relax

```

```
\cglformat{##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\@cGlsp@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cGlspformat{##2}{##3}%
\glsunset{##2}%
\else
\@Glsp@{##1}{##2}[##3]%
\fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*\@gls@entry@count}[2]{%
\csedef{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*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{%
  \@gls@default@entryfmt\glsdisplayfirst\glsdisplay
}
\newcommand*{\@gls@default@entryfmt}[2]{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          #2{\glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
        {%
          #1{\glsentryfirstplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \ifbool{glscompatible-3.07}%
        {%
          \protected@edef\@glo@etext{%
            #2{\glsentryplural{\glslabel}}%
            {\glsentrydescplural{\glslabel}}%
            {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
          \xmakefirstuc\@glo@etext
        }%
        {%
          #2{\Glsentryplural{\glslabel}}%
          {\glsentrydescplural{\glslabel}}%
          {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
        }%
      }%
    }%
  }%

```

```

{%
  \ifbool{glscompatible-3.07}%
  {%
    \protected@edef\@glo@etext{%
      #1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
    \xmakefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirstplural{\glslabel}}%
    {\glsentrydescplural{\glslabel}}%
    {\glsentrysymbolplural{\glslabel}}{\glsinsert}%
  }%
}%
}%
{%
  \ifglsused\glslabel
  {%
    \mfirstucMakeUppercase{#2{\glsentryplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
  {%
    \mfirstucMakeUppercase{#1{\glsentryfirstplural{\glslabel}}%
      {\glsentrydescplural{\glslabel}}%
      {\glsentrysymbolplural{\glslabel}}{\glsinsert}}%
  }%
}%
}%
{%
  \glscapscase
  {%
    \ifglsused\glslabel
    {%
      #2{\glsentrytext{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
    {%
      #1{\glsentryfirst{\glslabel}}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{\glsinsert}%
    }%
  }%
}%
{%
  \ifglsused\glslabel
  {%
    \ifbool{glscompatible-3.07}%
    {%

```

```

\protected@edef\@glo@etext{%
  #2{\glsentrytext{\glslabel}}%
  {\glsentrydesc{\glslabel}}%
  {\glsentrysymbol{\glslabel}}{\glsinsert}}%
\makefirstuc\@glo@etext
}%
{%
  #2{\Glsentrytext{\glslabel}}%
  {\glsentrydesc{\glslabel}}%
  {\glsentrysymbol{\glslabel}}{\glsinsert}%
}%
}%
{%
\ifbool{glscompatible-3.07}%
{%
  \protected@edef\@glo@etext{%
    #1{\glsentryfirst{\glslabel}}%
    {\glsentrydesc{\glslabel}}%
    {\glsentrysymbol{\glslabel}}{\glsinsert}}%
    \makefirstuc\@glo@etext
  }%
  {%
    #1{\Glsentryfirst{\glslabel}}%
    {\glsentrydesc{\glslabel}}%
    {\glsentrysymbol{\glslabel}}{\glsinsert}%
  }%
}%
}%
}%
\ifglsused\glslabel
{%
  \mfirstucMakeUppercase{#2{\glsentrytext{\glslabel}}%
    {\glsentrydesc{\glslabel}}%
    {\glsentrysymbol{\glslabel}}{\glsinsert}}%
}%
{%
  \mfirstucMakeUppercase{#1{\glsentryfirst{\glslabel}}%
    {\glsentrydesc{\glslabel}}%
    {\glsentrysymbol{\glslabel}}{\glsinsert}}%
}%
}%
}%
}%
\ifglsused{\glslabel}%
{%
  #2{\glscustomtext}%
  {\glsentrydesc{\glslabel}}%
  {\glsentrysymbol{\glslabel}}{\glsinsert}%
}%

```

```

    {%
      #1{\glscustomtext}%
      {\glsentrydesc{\glslabel}}%
      {\glsentrysymbol{\glslabel}}{%}%
    }%
  }%
}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glusifplural
    {%
      \glscapscase
      {%
        \ifglssused\glslabel
        {%
          \glsentryplural{\glslabel}\glsininsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsininsert
        }%
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsininsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsininsert
      }%
    }%
    {%
      \ifglssused\glslabel
      {%
        \mfirstucMakeUppercase
        {\glsentryplural{\glslabel}\glsininsert}%
      }%
      {%
        \mfirstucMakeUppercase
        {\glsentryfirstplural{\glslabel}\glsininsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \ifglssused\glslabel
      {%
        \glsentrytext{\glslabel}\glsininsert
      }%
    }%
  }%
}

```

```

        {%
        \glsentryfirst{\glslabel}\glsinsert
        }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \Glsentrytext{\glslabel}\glsinsert
    }%
    {%
    \Glsentryfirst{\glslabel}\glsinsert
    }%
    }%
    {%
    \ifglsused\glslabel
    {%
    \mfirstucMakeUppercase{\glsentrytext{\glslabel}\glsinsert}%
    }%
    {%
    \mfirstucMakeUppercase{\glsentryfirst{\glslabel}\glsinsert}%
    }%
    }%
    }%
    {%
    \glscustomtext\glsinsert
    }%
}
\newcommand*{\glsgenacfmt}{%
\ifdefempty\glscustomtext
{%
\ifglsused\glslabel
{%
\glsifplural
{%
\glscapscase
{%
\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
}%
{%
\mfirstucMakeUppercase
{\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
}%
}%
}%
\glscapscase
{%

```



```

        \acronymfont{\glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \mfirstucMakeUppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
}%
}%
}%
    \glscustomtext
}%
}
\newcommand*{\genacrfullformat}[2]{%
    \glsentrylong{#1}#2\space
    (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%

```

```

\protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
\xmakefirstuc\gls@text
}
\newcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \xmakefirstuc\gls@text
}
\newcommand*{\glsdisplayfirst}[4]{#1#4}
\newcommand*{\glsdisplay}[4]{#1#4}
\newcommand*{\defglsdisplay}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplay\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@display\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@displayfirst}}%
    {%
      \noexpand\@gls@default@entryfmt
      {\noexpand\csuse{gls@#1@displayfirst}}}%
    {\noexpand\csuse{gls@#1@display}}}%
  }%
  {%
    \noexpand\@gls@default@entryfmt
    {\noexpand\glsdisplayfirst}%
    {\noexpand\csuse{gls@#1@display}}}%
  }%
}%
\@gls@doentrydef
}
\newcommand*{\defglsdisplayfirst}[2][\glsdefaulttype]{%
  \GlossariesWarning{\string\defglsdisplayfirst\space is now obsolete.^^J
  Use \string\defglsentryfmt\space instead}%
  \expandafter\def\csname gls@#1@displayfirst\endcsname##1##2##3##4{#2}%
  \edef\@gls@doentrydef{%
    \noexpand\defglsentryfmt[#1]{%
      \noexpand\ifcsdef{gls@#1@display}}%
    {%
      \noexpand\@gls@default@entryfmt
      {\noexpand\csuse{gls@#1@displayfirst}}}%
    {\noexpand\csuse{gls@#1@display}}}%
  }%
  {%
    \noexpand\@gls@default@entryfmt
    {\noexpand\csuse{gls@#1@displayfirst}}}%
    {\noexpand\glsdisplay}%
  }%
}

```

```

    }%
  }%
}%
\@gls@doentrydef
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
{%
  \def\@gls@counter{#1}%
}%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\define@boolkey{glslink}{hyper}[true]{}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\define@boolkey{glslink}{local}[true]{}
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\glsifhyper}[2]{%
  \glslinkvar{#1}{#2}{#1}%
  \GlossariesWarning{\string\glsifhyper\space is deprecated. Did
    you mean \string\glsifhyperon\space or \string\glslinkvar?}%
}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3][]{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }
}

```

```

    }{%
      \glstextformat{#3}%
    }%
    \glspostlinkhook
  }
  \newcommand*{\glspostlinkhook}{}
  \newcommand*{@gls@link@checkfirsthyper}{%
    \ifglsused{\glslabel}%
    {%
    }%
    {%
      \gls@checkisacronymlist\glstype
      \ifglshyperfirst
        \if@glsisacronymlist
          \ifglsacrfootnote
            \KV@glslink@hyperfalse
          \fi
        \fi
      \else
        \KV@glslink@hyperfalse
      \fi
    }%
    \glslinkcheckfirsthyperhook
  }
  \newcommand*{\glslinkcheckfirsthyperhook}{}
  \newcommand*{\glslinkpostsetkeys}{}
  \newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
  \newcommand*{\do@glsdisablehyperinlist}{%
    \expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
    {\KV@glslink@hyperfalse}}%
  }
  \newcommand*{@gls@setdefault@glslink@opts}{}
  \def\@gls@link[#1]#2#3{%
    \leavevmode
    \protected@edef\glslabel{\glsdetoklabel{#2}}%
    \def\@gls@link@opts{#1}%
    \let\@gls@link@label\glslabel
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
    \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
    \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
    \@gls@setdefault@glslink@opts
    \do@glsdisablehyperinlist
    \do@gls@link@checkfirsthyper
    \setkeys{glslink}{#1}%
    \glslinkpostsetkeys
    \@gls@saveentrycounter
    \@gls@setsort{\glslabel}%
    \@do@wrglossary{#2}%
    \ifKV@glslink@hyper

```

```

\@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\else
\glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
\fi
\let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
\def\@gls@Hcounter{}%
\ifthenelse{\equal{\@gls@counter}{equation}}%
{
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglsentrycounter}%
}%
{%
\def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
}%
\protected@edef\theHglentrycounter{\@gls@Hcounter}%
\restorecounters@
\fi
}%
}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
\ifx\@gls@Hcounter\@empty
\ifcsundef{theH\@gls@counter}%
{%
\def\theHglentrycounter{\theglsentrycounter}%
}%
{%
\protected@edef\theHglentrycounter{\expandafter\noexpand
\csname theH\@gls@counter\endcsname}%
}%
}

```

```

\fi
}
\def\@set@glo@numformat#1#2#3#4{%
\expandafter\@glo@check@mkidxrangechar#3\@nil
\protected@edef#1{%
\@glo@prefix setentrycounter[#4]{#2}%
\expandafter\string\csname\@glo@suffix\endcsname
}%
\@gls@checkmkidxchars#1%
}
\def\@glo@check@mkidxrangechar#1#2\@nil{%
\if#1(\relax
\def\@glo@prefix{)%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\if#1)\relax
\def\@glo@prefix{)}%
\if\relax#2\relax
\def\@glo@suffix{glsnumberformat}%
\else
\def\@glo@suffix{#2}%
\fi
\else
\def\@glo@prefix{}\def\@glo@suffix{#1#2}%
\fi
\fi}
\newcommand*{\@gls@escbsdq}[1]{%
\def\@gls@checkedmkidx{}%
\let\gls@xdystring=#1\relax
\@onelevel@sanitize\gls@xdystring
\edef\do@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash\gls@xdystring\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null}%
\do@gls@xdycheckbackslash
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@xdycheckquote\gls@xdystring\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{\gls@xdystring}%
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dostubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
}

```

```

\gls@dostsubst
}%
\let#1=\gls@xdyststring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifglsxindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescactual#1\@nil"??"\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkescbar#1\@nil\\|\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\expandafter\@gls@checkesclevel#1\@nil\!\!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{%
\edef\@do@gls@checkesc{%
\noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
}%
\do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkquote{\relax}%

```

```

\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
\def\@@gls@checkquote{\@gls@checkquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkquote{\@gls@checkquote#2""\null}%
\else
\def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
\fi
\fi
\@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@@gls@checkescquote{\@gls@checkescquote#2""\null}%
\else
\def\@@gls@checkescquote{\@gls@checkescquote#2"#3\null}%
\fi
\fi
\@@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar

```



```

        \@gls@quotechar\string\" \@gls@actualchar}%
        \def\@gls@checkescactual{\@gls@checkescactual#3\null}%
    \fi
\else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
    \@gls@quotechar\string\" \@gls@actualchar}%
    \ifx\null#3\null
        \def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
    \else
        \def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
    \fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkescbar{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string\" \@gls@encapchar
            \@gls@quotechar\string\" \@gls@encapchar}%
            \def\@gls@checkescbar{\@gls@checkescbar#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\string\" \@gls@encapchar}%
        \ifx\null#3\null
            \def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
        \else
            \def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
        \fi
    \fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@checkesclevel{\relax}%
        \else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotechar\string\" \@gls@levelchar
            \@gls@quotechar\string\" \@gls@levelchar}%
            \def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
        \fi
    \fi
\@gls@checkesclevel
}

```

```

\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string\"@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%

```

```

\ifx\null#3\null
  \def\@gls@checklevel{\@gls@checklevel#2!\null}%
\else
  \def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkactual{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%
      \def\@gls@checkactual{\@gls@checkactual#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@actualchar}%
    \ifx\null#3\null
      \def\@gls@checkactual{\@gls@checkactual#2??\null}%
    \else
      \def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
    \fi
  \fi
  \@gls@checkactual
}
\bgroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
  \gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@gls@checkesc{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
      \def\@gls@checkesc{\@gls@checkesc#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@backslashchar}%
    \ifx\null#3\null

```

```

        |def|@gls@checkesc{|@gls@checkesc#2\\|null}%
    |else
        |def|@gls@checkesc{|@gls@checkesc#2\#3|null}%
    |fi
|fi
|@gls@checkesc
}
|egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
    \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
    \toks@={#1}%
    \ifx\null#2\null
        \ifx\null#3\null
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
            \def\@gls@xdycheckquote{\relax}%
        |else
            \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
                \string"\string"}%
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
        \fi
    |else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \string"}%
        \ifx\null#3\null
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
        |else
            \def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
        \fi
    \fi
    \@gls@xdycheckquote
}
\edef\def\@gls@xdycheckbackslash{%
    \noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
        ##2\@backslashchar##3\noexpand\null{%
        \noexpand\@gls@tmpb=\noexpand\expandafter
            {\noexpand\@gls@checkedmkidx}%
        \noexpand\toks@={##1}%
        \noexpand\ifx\noexpand\null##2\noexpand\null
            \noexpand\ifx\noexpand\null##3\noexpand\null
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
                \noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
            \noexpand|else
                \noexpand\edef\noexpand\@gls@checkedmkidx{%
                    \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
                    \@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
            \noexpand\def\noexpand\@gls@xdycheckbackslash{%
                \noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
            \noexpand\fi
        \noexpand|else

```

```

\noexpand\edef\noexpand\@gls@checkedmkidx{%
  \noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
  \@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    \@backslashchar\noexpand\null}%
\noexpand\else
  \noexpand\def\noexpand\@gls@xdycheckbackslash{%
    \noexpand\@gls@xdycheckbackslash##2\@backslashchar
    ##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

\newlength@gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
  \glsshowtarget{#1}%
  \settoheight{\gls@tmplen}{#2}%
  \raisebox{\gls@tmplen}{\hypertarget{#1}{}}#2%
}
\newcommand*{\glsdohyperlink}[2]{%
  \glsshowtarget{#1}%
  \hyperlink{#1}{#2}%
}
\newcommand*{\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue

```

```

\let\@glslink\glsdohyperlink
\let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\newcommand*{\@Gls}[2][]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \ifKV@glslink@local
      \glslocalunset{#2}%
    \else
      \glsunset{#2}%
    \fi
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}

```

```

\newcommand*{\@GLS}[2][]{%
  \new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}%
}
\def\@GLS@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glS@link@checkfirsthyper\@glS@link@checkfirsthyper
    \let\glSifplural\@secondoftwo
    \let\glScapscase\@thirdofthree
    \let\glScustomtext\@empty
    \def\glSinsert{#3}%
    \def\@glo@text{\csname glS@\glstype @entryfmt\endcsname}%
    \@glS@link[#1]{#2}{\@glo@text}%
    \ifKV@glSlink@local
      \glSlocalunset{#2}%
    \else
      \glSunset{#2}%
    \fi
  }%
  \glSpostlinkhook
}
\newrobustcmd*{\glSpl}{\@glS@hyp@opt\@glSpl}
\newcommand*{\@glSpl}[2][]{%
  \new@ifnextchar[{\@glSpl@{#1}{#2}}{\@glSpl@{#1}{#2}[]}%
}
\def\@glSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glS@link@checkfirsthyper\@glS@link@checkfirsthyper
    \let\glSifplural\@firstoftwo
    \let\glScapscase\@firstofthree
    \let\glScustomtext\@empty
    \def\glSinsert{#3}%
    \def\@glo@text{\csname glS@\glstype @entryfmt\endcsname}%
    \@glS@link[#1]{#2}{\@glo@text}%
    \ifKV@glSlink@local
      \glSlocalunset{#2}%
    \else
      \glSunset{#2}%
    \fi
  }%
  \glSpostlinkhook
}
\newrobustcmd*{\GlSpl}{\@glS@hyp@opt\@GlSpl}
\newcommand*{\@GlSpl}[2][]{%
  \new@ifnextchar[{\@GlSpl@{#1}{#2}}{\@GlSpl@{#1}{#2}[]}%
}
\def\@GlSpl@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%

```

```

\let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
\let\gl@sifplural\@firstoftwo
\let\gl@scapscase\@secondofthree
\let\glscustomtext\@empty
\def\gl$insert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gl@link[#1]{#2}{\@glo@text}%
\ifKV@gl@link@local
  \gl@localunset{#2}%
\else
  \gl@unset{#2}%
\fi
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLSp1}{\@gl@hyp@opt\@GLSp1}
\newcommand*{\@GLSp1}[2][{}]{%
  \new@ifnextchar[{\@GLSp1@{#1}{#2}}{\@GLSp1@{#1}{#2}[]}%
}
\def\@GLSp1@#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\glscustomtext\@empty
    \def\gl$insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else
      \gl@unset{#2}%
    \fi
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@doifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\glscustomtext{#3}%
    \def\gl$insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \ifKV@gl@link@local
      \gl@localunset{#2}%
    \else

```



```

        \glsunset{#2}%
    \fi
}%
\glspostlinkhook
}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \@gls@link[#1]{#2}{#3}%
    }%
    \glspostlinkhook
}
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
\newcommand*{\@glstext}[2][]{%
    \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
\def\@glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glstentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\newcommand*{\@GLStext}[2][]{%
    \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
\def\@GLStext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\newcommand*{\@Glstext}[2][]{%
    \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
\def\@Glstext@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glstentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][]{%
    \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
\def\@glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\glstentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\newcommand*{\@Glsfirst}[2][]{%
    \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}}
\def\@Glsfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\newcommand*{\@GLSfirst}[2][]{%
    \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]}}
\def\@GLSfirst@#1#2[#3]{%
    \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glstentryfirst{#2}#3}}%
}
}

```

```

\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} [] }}
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} [] }}
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} [] }}
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} [] }}
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} [] }}
\def\@Glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\GLSfirstplural}{\@gls@hyp@opt\@GLSfirstplural}
\newcommand*{\@GLSfirstplural}[2] [] {%
  \new@ifnextchar[{\@GLSfirstplural@{#1}{#2}}{\@GLSfirstplural@{#1}{#2} [] }}
\def\@GLSfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\glsname}{\@gls@hyp@opt\@glsname}
\newcommand*{\@glsname}[2] [] {%
  \new@ifnextchar[{\@glsname@{#1}{#2}}{\@glsname@{#1}{#2} [] }}
\def\@glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryname{#2}#3}%
}
\newrobustcmd*{\Glsname}{\@gls@hyp@opt\@Glsname}
\newcommand*{\@Glsname}[2] [] {%
  \new@ifnextchar[{\@Glsname@{#1}{#2}}{\@Glsname@{#1}{#2} [] }}
\def\@Glsname@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryname{#2}#3}%
}
\newrobustcmd*{\GLSname}{\@gls@hyp@opt\@GLSname}
\newcommand*{\@GLSname}[2] [] {%

```

```

\new@ifnextchar [{\@GLSname@{#1}{#2}}{\@GLSname@{#1}{#2} []}}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryname{#2}#3}}%
}
\newrobustcmd*{\@glsdesc}{\@gls@hyp@opt\@glsdesc}
\newcommand*{\@glsdesc}[2] [] {%
  \new@ifnextchar [{\@glsdesc@{#1}{#2}}{\@glsdesc@{#1}{#2} []}}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydesc{#2}#3}%
}
\newrobustcmd*{\@GLSdesc}{\@gls@hyp@opt\@GLSdesc}
\newcommand*{\@GLSdesc}[2] [] {%
  \new@ifnextchar [{\@GLSdesc@{#1}{#2}}{\@GLSdesc@{#1}{#2} []}}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydesc{#2}#3}}%
}
\newrobustcmd*{\@glsdescplural}{\@gls@hyp@opt\@glsdescplural}
\newcommand*{\@glsdescplural}[2] [] {%
  \new@ifnextchar [{\@glsdescplural@{#1}{#2}}{\@glsdescplural@{#1}{#2} []}}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\@GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrydescplural{#2}#3}%
}
\newrobustcmd*{\@GLSdescplural}{\@gls@hyp@opt\@GLSdescplural}
\newcommand*{\@GLSdescplural}[2] [] {%
  \new@ifnextchar [{\@GLSdescplural@{#1}{#2}}{\@GLSdescplural@{#1}{#2} []}}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentrydescplural{#2}#3}}%
}
\newrobustcmd*{\@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*{\glssymbolplural}{\@gls@hyp@opt\@glssymbolplural}
\newcommand*{\@glssymbolplural}[2][]{%
  \new@ifnextchar[{\@glssymbolplural@{#1}{#2}}{\@glssymbolplural@{#1}{#2}[]}}
\def\@glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\Glssymbolplural}{\@gls@hyp@opt\@Glssymbolplural}
\newcommand*{\@Glssymbolplural}[2][]{%
  \new@ifnextchar[{\@Glssymbolplural@{#1}{#2}}{\@Glssymbolplural@{#1}{#2}[]}}
\def\@Glssymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrysymbolplural{#2}#3}%
}
\newrobustcmd*{\GLSsymbolplural}{\@gls@hyp@opt\@GLSsymbolplural}
\newcommand*{\@GLSsymbolplural}[2][]{%
  \new@ifnextchar[{\@GLSsymbolplural@{#1}{#2}}{\@GLSsymbolplural@{#1}{#2}[]}}
\def\@GLSsymbolplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentrysymbolplural{#2}#3}}%
}
\newrobustcmd*{\glsuseri}{\@gls@hyp@opt\@glsuseri}
\newcommand*{\@glsuseri}[2][]{%
  \new@ifnextchar[{\@glsuseri@{#1}{#2}}{\@glsuseri@{#1}{#2}[]}}
\def\@glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\Glsuseri}{\@gls@hyp@opt\@Glsuseri}
\newcommand*{\@Glsuseri}[2][]{%
  \new@ifnextchar[{\@Glsuseri@{#1}{#2}}{\@Glsuseri@{#1}{#2}[]}}
\def\@Glsuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuseri{#2}#3}%
}
\newrobustcmd*{\GLSuseri}{\@gls@hyp@opt\@GLSuseri}
\newcommand*{\@GLSuseri}[2][]{%
  \new@ifnextchar[{\@GLSuseri@{#1}{#2}}{\@GLSuseri@{#1}{#2}[]}}
\def\@GLSuseri@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\Glsentryuseri{#2}#3}}%
}
\newrobustcmd*{\glsuserii}{\@gls@hyp@opt\@glsuserii}
\newcommand*{\@glsuserii}[2][]{%
  \new@ifnextchar[{\@glsuserii@{#1}{#2}}{\@glsuserii@{#1}{#2}[]}}
\def\@glsuserii@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryuserii{#2}#3}%
}

```



```

\new@ifnextchar [{\@glsuserv@{#1}{#2}}{\@glsuserv@{#1}{#2} []}}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\Glsuserv}{\@gls@hyp@opt\@Glsuserv}
\newcommand*{\@Glsuserv}[2] [] {%
  \new@ifnextchar [{\@Glsuserv@{#1}{#2}}{\@Glsuserv@{#1}{#2} []}}
\def\@Glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuserv{#2}#3}%
}
\newrobustcmd*{\GLSuserv}{\@gls@hyp@opt\@GLSuserv}
\newcommand*{\@GLSuserv}[2] [] {%
  \new@ifnextchar [{\@GLSuserv@{#1}{#2}}{\@GLSuserv@{#1}{#2} []}}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuserv{#2}#3}}%
}
\newrobustcmd*{\glsuservi}{\@gls@hyp@opt\@glsuservi}
\newcommand*{\@glsuservi}[2] [] {%
  \new@ifnextchar [{\@glsuservi@{#1}{#2}}{\@glsuservi@{#1}{#2} []}}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\Glsuservi}{\@gls@hyp@opt\@Glsuservi}
\newcommand*{\@Glsuservi}[2] [] {%
  \new@ifnextchar [{\@Glsuservi@{#1}{#2}}{\@Glsuservi@{#1}{#2} []}}
\def\@Glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryuservi{#2}#3}%
}
\newrobustcmd*{\GLSuservi}{\@gls@hyp@opt\@GLSuservi}
\newcommand*{\@GLSuservi}[2] [] {%
  \new@ifnextchar [{\@GLSuservi@{#1}{#2}}{\@GLSuservi@{#1}{#2} []}}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\mfirstucMakeUppercase{\glsentryuservi{#2}#3}}%
}
\newrobustcmd*{\acrshort}{\@gls@hyp@opt\@ns@acrshort}
\newcommand*{\@ns@acrshort}[2] [] {%
  \new@ifnextchar [{\@acrshort{#1}{#2}}{\@acrshort{#1}{#2} []}}
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsentryshort{#2}}#3%
    }%
    \@gls@link{#1}{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
}

```

```

\glspostlinkhook
}
\newrobustcmd*{\Acrshort}{\@gls@hyp@opt\@ns@Acrshort}
\newcommand*{\ns@Acrshort}[2] [] {%
  \new@ifnextchar[{\@Acrshort{#1}{#2}}{\@Acrshort{#1}{#2} []}]%
}
\def\@Acrshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\newcommand*{\ns@ACRshort}[2] [] {%
  \new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}]%
}
\def\@ACRshort#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\Glsentryshort{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gls@hyp@opt\@ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] [] {%
  \new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}]%
}
\def\@acrshortpl#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo

```

```

\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2][]{%
  \new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2}[]}%
}
\def\@Acrshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][]{%
  \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}%
}
\def\@ACRshortpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \mfirstucMakeUppercase{\acronymfont{\glentryshortpl{#2}}#3}%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
\newcommand*{\ns@acrlong}[2][]{%
  \new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2}[]}%
}

```



```

}
\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@firstofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \glsl@entrylong{#2}#3%
    }%
    \@glsl@link[#1]{#2}{\csname glsl@glstype @entryfmt\endcsname}%
  }%
  \glsl@postlinkhook
}
\newrobustcmd*{\Acrlong}{\@glsl@hyp@opt\@ns@Acrlong}
\newcommand*{\ns@Acrlong}[2][{}]{%
  \new@ifnextchar[{\@Acrlong{#1}{#2}}{\@Acrlong{#1}{#2}[]}%
}
\def\@Acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@secondofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \glsl@entrylong{#2}#3%
    }%
    \@glsl@link[#1]{#2}{\csname glsl@glstype @entryfmt\endcsname}%
  }%
  \glsl@postlinkhook
}
\newrobustcmd*{\ACrlong}{\@glsl@hyp@opt\@ns@ACrlong}
\newcommand*{\ns@ACrlong}[2][{}]{%
  \new@ifnextchar[{\@ACrlong{#1}{#2}}{\@ACrlong{#1}{#2}[]}%
}
\def\@ACrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@glsl@link@checkfirsthyper\@glsl@link@nocheckfirsthyper
    \def\glsl@label{#2}%
    \let\glsl@ifplural\@secondoftwo
    \let\glsl@capscase\@thirdofthree
    \let\glsl@insert\@empty
    \def\glsl@customtext{%
      \mfirstucMakeUppercase{\glsl@entrylong{#2}#3}%
    }%
  }%
}

```

```

    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gls@hyp@opt\ns@acrlongpl}
\newcommand*{\ns@acrlongpl}[2][]{%
  \new@ifnextchar[{\@acrlongpl{#1}{#2}}{\@acrlongpl{#1}{#2}[]}%
}
\def\@acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glstrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\AcrLongpl}{\@gls@hyp@opt\ns@AcrLongpl}
\newcommand*{\ns@AcrLongpl}[2][]{%
  \new@ifnextchar[{\@AcrLongpl{#1}{#2}}{\@AcrLongpl{#1}{#2}[]}%
}
\def\@AcrLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRLongpl}{\@gls@hyp@opt\ns@ACRLongpl}
\newcommand*{\ns@ACRLongpl}[2][]{%
  \new@ifnextchar[{\@ACRLongpl{#1}{#2}}{\@ACRLongpl{#1}{#2}[]}%
}
\def\@ACRLongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper

```

```

\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \mfirstucMakeUppercase{\glsentrylongpl{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
  \csname glo@\glsdetoklabel{#1}@\#2\endcsname
}
\newcommand*{\glsletentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@\#3}%
}
\newcommand*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@\#2}%
    \ifdef\@glo@text
    {%
      \xmakefirstuc{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ??%
  }%
}
\newcommand*{\glsentryname}[1]{\@gls@entry@field{#1}{name}}
\newrobustcmd*{\Glsentryname}[1]{%
  \@Gls@entryname{#1}%
}
\newcommand*{\@Gls@entryname}[1]{%
  \@Gls@entry@field{#1}{name}%
}
\newcommand*{\@Gls@acentryname}[1]{%
  \ifglshaslong{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@name}%
    \expandafter\@gls@getbody\@glo@text{}\@nil
    \expandafter\ifx\@gls@body\glsentrylong\relax
    \expandafter\Glsentrylong\@gls@rest
  }%
  \else

```

```

\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\xmakefirstuc{\@glo@text}%
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*{\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newrobustcmd*{\Glsentrydesc}[1]{%
\@Gls@entry@field{#1}{desc}%
}
\newcommand*{\glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newrobustcmd*{\Glsentrydescplural}[1]{%
\@Gls@entry@field{#1}{descplural}%
}
\newcommand*{\glsentrytext}[1]{\@Gls@entry@field{#1}{text}}
\newrobustcmd*{\Glsentrytext}[1]{%
\@Gls@entry@field{#1}{text}%
}
\newcommand*{\glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newrobustcmd*{\Glsentryplural}[1]{%
\@Gls@entry@field{#1}{plural}%
}
\newcommand*{\glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}
\newrobustcmd*{\Glsentrysymbol}[1]{%
\@Gls@entry@field{#1}{symbol}%
}

```

```

}
\newcommand*{\glsentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newrobustcmd*{\Glsentrysymbolplural}[1]{%
  \@Gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\glsentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newrobustcmd*{\Glsentryfirst}[1]{%
  \@Gls@entry@field{#1}{first}%
}
\newcommand*{\glsentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newrobustcmd*{\Glsentryfirstplural}[1]{%
  \@Gls@entry@field{#1}{firstpl}%
}
\newcommand*{\glscapitalisewords}[1]{%
  \@capitalisewords{#1}%
}
\newrobustcmd*{\@glsentrytitlecase}[2]{%
  \glsdoifexists{#1}%
  {%
    \glsfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\ifdef\texorpdfstring
{
  \newcommand*{\glsentrytitlecase}[2]{%
    \texorpdfstring
      {\@glsentrytitlecase{#1}{#2}}%
      {\@gls@entry@field{#1}{#2}}%
  }
}
{
  \newcommand*{\glsentrytitlecase}[2]{\@glsentrytitlecase{#1}{#2}}
}
\newcommand*{\glsentrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glsentrysort}[1]{%
  \@gls@entry@field{#1}{sort}%
}
\newcommand*{\glsentryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glsentryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}

```

```

\newrobustcmd*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\glsentryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newrobustcmd*{\Glsentryuserii}[1]{%
  \@Gls@entry@field{#1}{userii}%
}
\newcommand*{\glsentryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newrobustcmd*{\Glsentryuseriii}[1]{%
  \@Gls@entry@field{#1}{useriii}%
}
\newcommand*{\glsentryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newrobustcmd*{\Glsentryuseriv}[1]{%
  \@Gls@entry@field{#1}{useriv}%
}
\newcommand*{\glsentryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}
\newrobustcmd*{\Glsentryuserv}[1]{%
  \@Gls@entry@field{#1}{userv}%
}
\newcommand*{\glsentryuservi}[1]{%
  \@gls@entry@field{#1}{uservi}%
}
\newrobustcmd*{\Glsentryuservi}[1]{%
  \@Gls@entry@field{#1}{uservi}%
}
\newcommand*{\glsentryshort}[1]{\@gls@entry@field{#1}{short}}
\newrobustcmd*{\Glsentryshort}[1]{%
  \@Gls@entry@field{#1}{short}%
}
\newcommand*{\glsentryshortpl}[1]{\@gls@entry@field{#1}{shortpl}}
\newrobustcmd*{\Glsentryshortpl}[1]{%
  \@Gls@entry@field{#1}{shortpl}%
}
\newcommand*{\glsentrylong}[1]{\@gls@entry@field{#1}{long}}
\newrobustcmd*{\Glsentrylong}[1]{%
  \@Gls@entry@field{#1}{long}%
}
\newcommand*{\glsentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newrobustcmd*{\Glsentrylongpl}[1]{%
  \@Gls@entry@field{#1}{longpl}%
}
\newcommand*{\glsentryfull}[1]{%

```

```

\acrfullformat{\glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newrobustcmd*{\Glsentryfull}[1]{%
\acrfullformat{\Glsentrylong{#1}}{\acronymfont{\glsentryshort{#1}}}%
}
\newcommand*{\glsentryfullpl}[1]{%
\acrfullformat{\glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newrobustcmd*{\Glsentryfullpl}[1]{%
\acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\glsentryshortpl{#1}}}%
}
\newcommand*{\glsentrynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\@gls@entry@field{#1}{numberlist}%
}%
}
\@ifpackageloaded{hyperref}{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\GlossariesWarning
{%
\string\glsdisplaynumberlist\space
doesn't work with hyperref.^^JUsing
\string\glsentrynumberlist\space instead%
}%
\glsentrynumberlist{#1}%
}%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
\glsdoifexists{#1}%
{%
\bgroup
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\let\@org@glsnumberformat\glsnumberformat
\def\glsnumberformat##1{##1}%
\protected@edef\the@numberlist{%
\csname glo@\@glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{}%
\def\@gls@numlist@nextsep{}%
\def\@gls@numlist@lastsep{}%
\def\@gls@thislist{}%
\def\@gls@donext@def{}%
\renewcommand\do[1]{%
\protected@edef\@gls@thislist{%
\@gls@thislist
\noexpand\@gls@numlist@sep
##1%
}%
\let\@gls@numlist@sep\@gls@numlist@nextsep

```

```

\def\@gls@numlist@nextsep{\glsnumlistsep}%
\@gls@donext@def
\def\@gls@donext@def{%
\def\@gls@numlist@lastsep{\glsnumlistlastsep}%
}%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
\def\@glo@label{#2}%
\@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][]{%
\@gls@adjustmode
\glsdoifexists{#2}%
{%
\def\@glsnumberformat{glsnumberformat}%
\protected@edef\@gls@counter{\csname glo@%glsdetoklabel{#2}@counter\endcsname}%
\setkeys{glossadd}{#1}%
\@gls@saveentrycounter
\@gls@setsort{#2}%
\@do@wrglossary{#2}%
}%
}
\newcommand*{\@gls@adjustmode}{}
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}
\newrobustcmd*{\glsaddall}[1][]{%
\protected@edef\@glo@type{\@glo@types}%
\setkeys{glossadd}{#1}%
\forallglsentries[\@glo@type]{\@glo@entry}{%
\glsadd[#1]{\@glo@entry}%
}%
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
\forallglsentries[#1]{\@glo@entry}%
{%
\ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
}%
}
\newcommand*{\glsignore}[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{}}

```



```

\edef\glsclosebrace{\expandafter\@gobble\string\}}
\edef\glslbackslash{\expandafter\@gobble\string\}
\edef\glquote#1{\string"#1\string"}
\edef\glpercentchar{\expandafter\@gobble\string\%}
\edef\glstildechar{\string~}
\ifglxindy
  \newcommand*{\@glsfirstletter}{A}
\fi
\newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
  \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
\ifglxindy
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \renewcommand*{\@glsfirstletter}{#1}%
    \@onelevel@sanitize\@glsfirstletter
  }
\else
  \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
    \glsnxindywarning\GlsSetXdyFirstLetterAfterDigits}
  \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
    \@GlsSetXdyFirstLetterAfterDigits
  }
\fi
\ifglxindy
  \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
  \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder}
\ifglxindy
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
  }
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
    \renewcommand*{\@xdynumbergrouporder}{#1}%
    \@onelevel@sanitize\@xdynumbergrouporder
  }
\else
  \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
    \glsnxindywarning\GlsSetXdyNumberGroupOrder}
  \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
    \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglxindy
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
    \renewcommand*{\@glsminrange}{#1}}
\else
  \newcommand*{\GlsSetXdyMinRangeLength}[1]{%

```

```

\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglxindy
\def\writeist{%
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \@gls@addpredefinedattributes
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \@for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \@for\@gls@classI:=\@gls@xdy@locationlist\do{%
    \protected@write\glswrite{{(define-location-class
      \string"\@gls@classI\string"^^J\space\space\space
      (
        :sep "{{"
        \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
        :sep "}"
      )
      ^^J\space\space\space
      :min-range-length \@glsminrange^^J%
    )
  }%
  {%
    \@for\@gls@classII:=\@gls@xdy@locationlist\do{%
      \protected@write\glswrite{{(define-location-class
        \string"\@gls@classII-\@gls@classI\string"
        ^^J\space\space\space
        (
          :sep "{"
          \csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
          :sep "{{"
          \csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
          :sep "}"
        )
      )
    }%
  }%
}

```

```

        ^^J\space\space\space
        :min-range-length \@glxminrange^^J%
    )
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glssseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected\write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string
\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}%
\write\glswrite{(markup-letter-group-list

```

```

        :sep \string"\string\glsgroupskip\glstildechar n\string"))}%
\write\glswrite{(markup-indexentry
        :open \string"\string\relax \string\glsresetentrylist
        \glstildechar n\string"))}%
\write\glswrite{(markup-locclass-list :open
        \string"\glsopenbrace\string\glossaryentrynumbers
        \glsopenbrace\string\relax\space \string"^^J\space\space\space
        :sep \string", \string"
        :close \string"\glsclosebrace\glsclosebrace\string"))}%
\write\glswrite{(markup-locref-list
        :sep \string"\string\delimN\space\string"))}%
\write\glswrite{(markup-range
        :sep \string"\string\delimR\space\string"))}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
        \write\glswrite{(markup-range
                :close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
        :sep \string"\string\glsgroupskip\glstildechar n\string"))}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
        :open-head \string"\string\glsgroupheading
        \glsopenbrace\string"^^J\space\space\space
        :close-head \string"\glsclosebrace\string"))}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}%
\let\GlsSetQuote\gls@nosetquote
\def\writeist{\relax

```

```

\ifundef{\glswrite}{\newwrite\glswrite}{\relax
\openout\glswrite=\istfilename
\write\glswrite{\glpercentchar\space makeindex style file
  created by the glossaries package}
\write\glswrite{\glpercentchar\space for document
  'jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '@glactualchar'}
\write\glswrite{encap '@glencapchar'}
\write\glswrite{level '@gllevelchar'}
\write\glswrite{quote '@glquotechar'}
\write\glswrite{keyword \string\string\glossaryentry\string}
\write\glswrite{preamble \string\string\glossarysection[\string
  \glossarytoctitle]{\string\glossarytitle}\string
  \glossarypreamble\string\n\string\begin{theglossary}\string
  \glossaryheader\string\n\string}
\write\glswrite{postamble \string\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string}
\write\glswrite{group_skip \string\string\glsgroupskip\string\n
  \string}
\write\glswrite{item_0 \string\string%\string\n\string}
\write\glswrite{item_1 \string\string%\string\n\string}
\write\glswrite{item_2 \string\string%\string\n\string}
\write\glswrite{item_01 \string\string%\string\n\string}
\write\glswrite{item_x1
  \string\string\relax \string\glresetentrylist\string\n
  \string}
\write\glswrite{item_12 \string\string%\string\n\string}
\write\glswrite{item_x2
  \string\string\relax \string\glresetentrylist\string\n
  \string}
\write\glswrite{delim_0 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_1 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_2 \string\string\{\string
  \glossaryentrynumbers\string\}\string\relax \string}
\write\glswrite{delim_t \string\string\}\string\}\string}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string\string\glsgroupheading\string\}\string}
\write\glswrite{heading_suffix
  \string\string\}\string\relax
  \string\glresetentrylist \string}
\write\glswrite{symhead_positive \string\glssymbols\string}
\write\glswrite{numhead_positive \string\glnumbers\string}
\write\glswrite{page_compositor \string\glscpositor\string}
\@glscbsdq\glscsuffixF

```

```

\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{ }
\ifglxsindy
\newcommand*\GlsSetQuote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\newcommand*\gls@nosetquote}[1]{\glsnomakeindexwarning\GlsSetQuote}
\else
\newcommand*\GlsSetQuote}[1]{\edef\@gls@quotechar{\string#1}%
\ifpackageloaded{tracklang}%
{%
\IfTrackedLanguage{german}%
{%
\def\@gls@extramakeindexopts{-g}%
}%
}%
}%
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkquote####1#1####2#1####3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={####1}%
\noexpand\ifx\noexpand\null####2\noexpand\null
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%

```

```

\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotearch\noexpand\@gls@quotearch}%
\noexpand\ifx\noexpand\null####3\noexpand\null
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1#1\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2#1####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\renewcommand\noexpand\@gls@checkmkidxchars}[1]{%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
#1#1\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
\noexpand||\noexpand||\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null

```

```

\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesclevel###1\noexpand\@nil
\noexpand\!\noexpand\!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesc###1\noexpand\@nil
\@backslashchar\@backslashchar\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{###1}%
}%
}%
\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
\noexpand\def\noexpand\@gls@checkescquote###1%
\expandonce{\csname#1\endcsname}###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={###1}%
\noexpand\ifx\noexpand\null###2\noexpand\null
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@checkescquote{\noexpand\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand|string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar
\noexpand\@gls@quotechar\noexpand|string\expandonce{%
\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\noexpand\@gls@quotechar\noexpand|string
\expandonce{\csname#1\endcsname}\noexpand\@gls@quotechar}%
\noexpand\ifx\noexpand\null###3\noexpand\null
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
\expandonce{\csname#1\endcsname}\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkescquote{%
\noexpand\@gls@checkescquote###2\expandonce{\csname#1\endcsname}%
###3\noexpand\null}%
\noexpand\fi

```



```

        \noexpand\fi
        \noexpand\@@gls@checkescquote
    }%
}%
\@gls@docheckquotedef
}
\newcommand*{\gls@nosetquote}[1]{\PackageError{glossaries}%
    {\string\GlsSetQuote\space not permitted here}%
    {Move \string\GlsSetQuote\space earlier in the preamble, as
    soon as possible after glossaries.sty has been loaded}}
\fi
\newcommand*{\@gls@extramakeindexopts}[1]{%
\newcommand{\noist}{%
    \@gls@addpredefinedattributes
    \let\writeist\relax
}
\newcommand*{\@makeglossary}[1]{%
    \ifglossaryexists{#1}%
    {%
        \ifglssavewrites
            \expandafter\newtoks\csname glo@#1@filetok\endcsname
        \else
            \expandafter\newwrite\csname glo@#1@file\endcsname
            \expandafter\@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
}
\ifglxsindy
\newcommand*{\@gls@automake}[1]{%
    \ifglossaryexists{#1}
    {%
        \@closegls{#1}%
        \ifdefstring{\glsorder}{letter}%
        {\def\@gls@order{-M ord/letorder }}%
        {\let\@gls@order\@empty}%
    }

```

```

\ifcsundef{@xdy@#1@language}%
{\let\@gls@langmod\@xdy@main@language}%
{\letcs\@gls@langmod{@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \@gls@istfilename\space
-C \@gls@codepage\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake}[1]{%
\ifglossaryexists{#1}
{%
\@closegls{#1}%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{@glotype@#1@log}
-o \jobname.\csuse{@glotype@#1@in}
\jobname.\csuse{@glotype@#1@out}}}%
}%
\@gls@dothiswrite
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-M ord/letorder }}%
{\let\@gls@order\@empty}%
\ifcsundef{@xdy@#1@language}%

```

```

        {\let\@gls@langmod\@xdy@main@language}%
        {\letcs\@gls@langmod{\@xdy@#1@language}}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{xindy
-I xindy
\@gls@order
-L \@gls@langmod\space
-M \gls@istfilebase\space
-C \gls@codepage\space
-t \jobname.\csuse{\@glotype@#1@log}
-o \jobname.\csuse{\@glotype@#1@in}
\jobname.\csuse{\@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{\@glotype@#1@out}}%
{%
\ifdefstring{\glsorder}{letter}%
{\def\@gls@order{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\immediate\noexpand\write18{makeindex \@gls@order
-s \istfilename\space
-t \jobname.\csuse{\@glotype@#1@log}
-o \jobname.\csuse{\@glotype@#1@in}
\jobname.\csuse{\@glotype@#1@out}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\@glotype@#1@out}
doesn't exist. Rerun may be required}}%
}%
{%
\GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\newcommand*{\@warn@nomakeglossaries}{}
\newcommand*{\warn@nomakeglossaries}{\@warn@nomakeglossaries}
\newcommand{\@gls@automake@immediate}{%
\ifnum\gls@automake@nr=2\relax

```

```

\@for\@gls@type:=\@glo@types\do{%
  \ifempty{\@gls@type}{}%
  {\@gls@automake@immediate{\@gls@type}}%
}%
\glsautomakefalse
\renewcommand*{\@gls@doautomake}{}%
\fi
}
\newcommand*{\makeglossaries}{%
\@domakeglossaries
{%
  \@gls@@automake@immediate
  \ifundef{\glswrite}{\newwrite\glswrite}{}%
  \protected@write\@auxout{}{\string\providecommand\string\@glsorder[1]{} }
  \protected@write\@auxout{}{\string\providecommand\string\@istfilename[1]{} }
  \ifundef\@gls@extramakeindexopts
  {}%
  {%
    \protected@write\@auxout{}{\string\providecommand
      \string\@gls@extramakeindexopts[1]{} }
    \protected@write\@auxout{}{\string\@gls@extramakeindexopts
      {\@gls@extramakeindexopts}}%
  }%
  \protected@write\@auxout{}{\string\@istfilename{\istfilename}}%
  \protected@write\@auxout{}{\string\@glsorder{\glsorder}}
  \@for\@glo@type:=\@glo@types\do{%
    \ifthenelse{\equal{\@glo@type}{}}{ }{%
      \@makeglossary{\@glo@type}}%
    }%
  \renewcommand*\newglossary[4][]{%
  \PackageError{glossaries}{New glossaries
must be created before \string\makeglossaries}{You need
to move \string\makeglossaries\space after all your
\string\newglossary\space commands}}%
  \let\@makeglossary\@gobble
  \let\makeglossaries\relax
  \@disable@onlypremakeg
  \let\gls@checkseeallowed\relax
  \let\warn@nomakeglossaries\relax
  \let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
  \iflssavenumberlist
    \edef\@gls@dodeflistparser{\noexpand\DeclareListParser
      {\noexpand\glsnumlistparser}{\delimN}}%
    \@gls@dodeflistparser
  \fi
  \let\makenoidxglossaries\@no@makeglossaries
  \renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
  }%
  \ifglsautomake

```

```

\renewcommand*{\@gls@doautomake}{%
  \@for\@gls@type:=\@glo@types\do{%
    \ifdefempty{\@gls@type}{}%
    {\@gls@automake{\@gls@type}}%
  }%
}%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}}%
      {%
        \string\glsseeformat##2}%
    }%
  }%
}

```

```

    }%
  }%
}%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{\glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{\glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
  }%

```

```

    }%
  }%
  \def\@gls@noidxloclist@finalsep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
  \@gls@noidxloclist@finalsep
  \@gls@noidxloclist@prev
}%
{%
  ??\glsdoifexists{##1}%
  {%
    \GlossariesWarning{Missing location list for ‘##1’. Either
      a rerun is required or you haven’t referenced the entry}%
  }%
}%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo\@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glsseeformat\glsseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[2]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}

```

```

\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can't use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
    {Either use one or other (or none) of those commands but not both
    together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {\string\glsnoidxdisplayloc
        {\@glo@counterprefix}%
        {\@gls@counter}%
        {\@glsnumberformat}%
        {\@glslocref}}%
      }%
  }%
}
\providecommand\istfile{\glswrite}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \foralllglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%

```



```

        \expandafter\the
        \csname glo@%glo@type @filetok\endcsname}%
    \immediate\closeout\glswrite
\fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
        \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
            {%
                \expandafter\protected@write\csname glo@#1@file\endcsname{%
                    \gls@disablepagerefexpansion}{#2}%
                }%
            {%
                \ifignoredglossary{#1}{}%
                {%
                    \GlossariesWarning{No file defined for glossary ‘#1’}%
                }%
            }%
        \fi
    \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \glswriteentry{#1}{\@do@wrglossary{#1}}%
}
\newcommand*{\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{\theHglentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \protected@edef\@do@gls@getcounterprefix{\noexpand\@gls@getcounterprefix
      {\@glslocref}{\@glsHlocref}%
    }%
    \@do@gls@getcounterprefix
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%
  \@@do@@wrglossary
}

```

```

\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodstrue
\newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
\beginingroup
\let\gls@orgthe\the
\let\gls@orgnumber\number
\let\gls@orgarabic\@arabic
\let\gls@orgromannumeral\romannumeral
\let\gls@orgalph\@alph
\let\gls@orgAlph\@Alph
\let\gls@orgRoman\@Roman
\ifglswrallowprimitivemods
\def\gls@the##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
\def\the{\expandafter\gls@the}%
\def\gls@number##1{%
\ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
\def\number{\expandafter\gls@number}%
\fi
\def\@arabic##1{%
\ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
\def\romannumeral##1{%
\ifx##1\c@page \gls@romanpage\else\gls@orgromannumeral##1\fi}%
\def\@Roman##1{%
\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theHglentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\theHglentrycounter\theHglentrycounter\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
\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)
:locoref \string"\@glo@counterprefix}{\@gls@locoref}\string" %
:attr \string"\@gls@counter\@glo@suffix\string"
\@glo@range
)
}%
\else
\@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@gls@numberformat}%
{\@glo@counterprefix}%
\gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
\string\glossaryentry{\csname glo@\@gls@label @index\endcsname
\@gls@encapchar\@glo@numfmt}{\@gls@locoref}}%
\fi
}
\newcommand*\@gls@getcounterprefix[2]{%
\edef\@gls@thisloc{#1}\edef\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been^^J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref

```

```

\ifglxsindy
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    (indexentry
      :tkey (\csname glo@#1@index\endcsname)
      :xref (\string"\@gls@xref\string")
      :attr \string"see\string"
    )
  }%
\else
  \gls@glossary{\csname glo@#1@type\endcsname}{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar glsseeformat\@gls@xref}{Z}}%
  \fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
  \ifx#2[\relax
    \@gls@fixbraces#1#2#3\@end@fixbraces
  \else
    \def#1{{#2#3}}%
  \fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
  \def#1{[#2]{#3}}%
}
\newrobustcmd*\glssee}[3][\seename]{%
  \@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
  \glssee[#1]{#3}{#2}}
\newrobustcmd*\glsseeformat}[3][\seename]{%
  \emph{#1} \glsseelist{#2}}
\newrobustcmd*\glsseelist}[1]{%
  \let\@gls@dolast\relax
  \let\@gls@donext\relax
  \@for\@gls@thislabel:=#1\do{%
    \ifx\@xfor@nextelement\@nnil
      \@gls@dolast
    \else
      \@gls@donext
    \fi
    \expandafter\glsseeitem\expandafter{\@gls@thislabel}%
    \let\@gls@dolast\glsseelastsep
    \let\@gls@donext\glsseesep
  }%
}
\newcommand*\glsseelastsep{\space\andname\space}
\newcommand*\glsseesep{, }
\newrobustcmd*\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*\glsseeitemformat}[1]{\glstrytext{#1}}
\newcommand*\gls@save@numberlist}[1]{%
  \ifglssavenumberlist

```

```

\toks@{#1}%
\protected@edef\do@writeaux@info{%
  \noexpand\csgdef{glo@glscurrententrylabel @numberlist}{\the\toks@}%
}%
\@onelevel@sanitize\do@writeaux@info
\protected@write\auxout{}\do@writeaux@info{%
\fi
}
\newcommand*\warn@noprntglossary{}{}%
\ifcsundef{printglossary}{}%
{%
  \@gls@warnonglossdefined
  \undef\printglossary
}
\newcommand*\@printgloss@checkexists@noignored}[2]{%
  \@ifglossaryexists{#1}%
  {#2}%
  {%
    \ifignoredglossary{#1}%
    {\GlossariesWarning{Glossary ‘#1’ is an ignored glossary}}%
    {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
  }%
}
\newcommand*\@printgloss@checkexists@allowignored}[2]{%
  \s@ifglossaryexists{#1}%
  {#2}%
  {\GlossariesWarning{Glossary ‘#1’ doesn’t exist}}%
}
\let\@printgloss@checkexists\@printgloss@checkexists@noignored
\newcommand*\@printglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*\@printglossaries{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*\@printnoidxglossary}[1][type=\glsdefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@noidx@glossary}%
}
\newcommand*\@printnoidxglossaries{%
  \forallglossaries{\@glo@type}{\printnoidxglossary[type=\@glo@type]}%
}
\newcommand*\@printgloss@setsort{}{}
\newcommand*\@gls@preglossaryhook{}{}
\newcommand*\@printglossary}[2]{%
  \def\@glo@type{\glsdefaulttype}%
  \def\glossarytitle{\csname @glotype@\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle

```

```

\def\@glossarystyle{%
  \ifx\@glossary@default@style\relax
    \GlossariesWarning{No default glossary style provided \MessageBreak
      for the glossary '\@glo@type'. \MessageBreak
      Using deprecated fallback. \MessageBreak
      To fix this set the style with \MessageBreak
      \string\setglossarystyle\space or use the \MessageBreak
      style key=value option}%
  \fi
}%
\def\gls@dotoc@title{\gls@set@toc@title{\@glo@type}}%
\let\@org@glossaryentrynumbers\glossaryentrynumbers
\bgroup
  \@printgloss@set@sort
  \setkeys{printgloss}{#1}%
  \@printgloss@check@exists{\@glo@type}%
  {%
    \ifx\glossarytitle\@org@glossarytitle
    \else
      \expandafter\let\csname @glo@type@\@glo@type @title\endcsname
        \glossarytitle
    \fi
    \let\currentglossary\@glo@type
    \let\org@glossaryentrynumbers\glossaryentrynumbers
    \let\gls@nonextpages\@gls@nonextpages
    \let\gls@nextpages\@gls@nextpages
    \let\@nopostdesc\@nopostdesc
    \gls@dotoc@title
    \@glossarystyle
    \let\gls@org@glossaryentryfield\glossentry
    \let\gls@org@glossarysubentryfield\subglossentry
    \renewcommand{\glossentry}[1]{%
      \protected@xdef\gls@currententrylabel{\gls@detoklabel{##1}}%
      \gls@org@glossaryentryfield{##1}%
    }%
    \renewcommand{\subglossentry}[2]{%
      \protected@xdef\gls@currententrylabel{\gls@detoklabel{##2}}%
      \gls@org@glossarysubentryfield{##1}{##2}%
    }%
    \@gls@preglossaryhook
    #2%
  }%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
  \makeatletter
  \@input@{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%
  \IfFileExists{\jobname.\csname @glo@type@\@glo@type @in\endcsname}%

```

```

}%
{\null}%
\ifglxindy
\ifcsundef{@xdy@\@glo@type @language}%
{%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\@xdy@main@language}}%
}%
}%
}%
{%
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@xdylanguage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@xdylanguage{\@glo@type}{\csname @xdy@\@glo@type
@language\endcsname}}%
}%
}%
}%
\@do@auxoutstuff
\edef\@do@auxoutstuff{%
\noexpand\AtEndDocument{%
\noexpand\immediate\noexpand\write\@auxout{%
\string\providecommand\string\@gls@codepage[2]{}}%
\noexpand\immediate\noexpand\write\@auxout{%
\string\@gls@codepage{\@glo@type}{\@gls@codepage}}%
}%
}%
\@do@auxoutstuff
\fi
\renewcommand*{\@warn@nomakeglossaries}{%
\GlossariesWarningNoLine{\string\makeglossaries\space
hasn't been used,^^Jthe glossaries will not be updated}%
}%
}
\newcommand*{\@glo@sortentries}[2]{%
\glosortentrieswarning
\def\@glo@sortinglist{}%
\def\@glo@sortinghandler{#1}%
\protected@edef\@glo@type{#2}%
\forlistcsloop{\@glo@do@sortentries}{\@glsref@#2}%
\csdef{\@glsref@#2}{}%
\@for\@this@label:=\@glo@sortinglist\do{%
\xifinlistcs{\@this@label}{\@glsref@#2}%

```



```

    {}%
    {%
        \listcsxadd{@glsref@#2}{\@this@label}%
    }%
    \ifcsdef{@glo@sortingchildren@\@this@label}%
    {%
        \@glo@addchildren{#2}{\@this@label}%
    }%
    {}%
}%
}
\newcommand*{\@glo@addchildren}[2]{%
    \bgroup
        \letcs{\@glo@childlist}{@glo@sortingchildren@#2}%
        \@for\@this@childlabel:=\@glo@childlist\do
        {%
            \xifinlistcs{\@this@childlabel}{@glsref@#1}%
            {}%
            {%
                \listcsxadd{@glsref@#1}{\@this@childlabel}%
            }%
            \ifcsdef{@glo@sortingchildren@\@this@childlabel}%
            {%
                \@glo@addchildren{#1}{\@this@childlabel}%
            }%
            {}%
            }%
        }%
    \egroup
}
\newcommand*{\@glo@do@sortentries}[1]{%
    \ifglshasparent{#1}%
    {%
        \protected@edef\@glo@parent{\csuse{glo@\glsdetoklabel{#1}@parent}}%
        \ifcsundef{@glo@sortingchildren@\@glo@parent}%
        {%
            \csdef{@glo@sortingchildren@\@glo@parent}{}%
        }%
        {}%
        \expandafter\@glo@sortedinsert
            \csname @glo@sortingchildren@\@glo@parent\endcsname{#1}%
        \xifinlistcs{\@glo@parent}{@glsref@\@glo@type}%
        {%
            }%
            {}%
            \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
        }%
    }%
    {%
        \@glo@sortedinsert{\@glo@sortinglist}{#1}%
    }%
}

```

```

}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
  \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
  \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
    {\expandonce\@gls@sort@B}%
    {\expandonce\@gls@sort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
}

```

```

    {%
      \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=word]}{%}%
    }%
  }
\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{%}%
  }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{%}%
  }%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{%}%
  }%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
}

```

```

{%
  \PackageError{glossaries}{Conflicting sort options:^^J
    \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
    \string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@\@glo@type}%
  {}%
  {%
    \listcsadd{\@glsref@\@glo@type}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
    \end{theglossary}%
  }%
}

```

```

\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{}}\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\ucode‘#1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@=‘#1\relax
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{\glo@\glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{\glo@\glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
{%
\subglossentry{\gls@level}{#1}%
{%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%

```

```

    }%
  }%
}%
{%
  \letcs{\@gls@sort}{glo@glstetoklabel{#1}@sort}%
  \expandafter\glo@grabfirst\@gls@sort{}\{}@nil
  \ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
  {}%
  {%
    \ifdefempty{\@gls@currentlettergroup}{}%
    {%
      \global\let\@glo@thislettergrp\@glo@thislettergrp
      \glsgroupskip
    }%
    \glsgroupheading{\@glo@thislettergrp}%
  }%
  \global\let\@gls@currentlettergroup\@glo@thislettergrp
  \ifdefvoid{\@gls@loclist}
  {%
    \glossentry{#1}{}%
  }%
  {%
    \glossentry{#1}%
    {%
      \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
    }%
  }%
}%
}
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%

```

```

\@gls@noidxloclist@sep
\@gls@noidxloclist@prev
\def\@gls@noidxloclist@prev{#1}%
}%
}
\newcommand*\glsnoidxdisplayloc[4]{%
\setentrycounter[#1]{#2}%
\csuse{#3}{#4}%
}
\newcommand*\@gls@reference}[3]{%
\glsdoifexistsorwarn{#2}%
{%
\ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
\ifinlistcs{#2}{\@glsref@#1}%
}%
{\listcsgadd{\@glsref@#1}{#2}%
\ifcsundef{glo@\glsdetoklabel{#2}@loclist}%
{\csgdef{glo@\glsdetoklabel{#2}@loclist}{}}%
}%
\listcsgadd{glo@\glsdetoklabel{#2}@loclist}{#3}%
}%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
\def\glossarytitle{#1}%
\let\gls@dotocitle\relax
}
\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dotocitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{\@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossentrycompatibility
\csname @glsstyle@#1\endcsname}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\@gls@numberedsection@nr\relax
\renewcommand*\@@glossarysecstar{*}%
\renewcommand*\@@glossaryseclabel{}%
\or

```

```

\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glsesubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
}
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
\gdef\glossaryentrynumbers##1{%
\glsresetentrylist
}%
}
\newcommand*{\@glsnextpages}{%
\gdef\glossaryentrynumbers##1{%
##1\glsresetentrylist}}

```



```

\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglsentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glsentrycounterlabel}{%
  \ifglsentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}

```

```

\newcommand*{\glsentryitem}[1]{%
  \ifglsentrycounter
    \glsstepentry{#1}\glsentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*{\glsesubentryitem}[1]{%
  \ifglsesubentrycounter
    \glsstepsubentry{#1}\glsesubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\providecommand*{\compatibleglossentry}[2]{%
  \toks@{#2}%
  \protected@edef\@do@Glossentry{\noexpand\glossaryentryfield{#1}%
    {\noexpand\glsnamefont
      {\expandafter\expandonce\csname glo@#1@name\endcsname}}}%
    {\expandafter\expandonce\csname glo@#1@desc\endcsname}}%
    {\expandafter\expandonce\csname glo@#1@symbol\endcsname}}%
    {\the\toks@}%
  }%
  \@do@Glossentry
}
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@#1\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }
}

```

```

    }%
  }
  \newcommand*{\Glossentrydesc}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrydesc{#1}%
    }%
  }
  \newcommand*{\glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \glsentrysymbol{#1}%
    }%
  }
  \newcommand*{\Glossentrysymbol}[1]{%
    \glsdoifexistsorwarn{#1}%
    {%
      \Glsentrysymbol{#1}%
    }%
  }
  \providecommand*{\compatiblesubglossentry}[3]{%
    \toks@{#3}%
    \protected@edef\@do@subglossentry{\noexpand\glossarysubentryfield{\number#1}%
    {#2}%
      {\noexpand\glsnamefont
        {\expandafter\expandonce\csname glo@#2@name\endcsname}}%
      {\expandafter\expandonce\csname glo@#2@desc\endcsname}%
      {\expandafter\expandonce\csname glo@#2@symbol\endcsname}%
      {\the\toks@}%
    }%
    \@do@subglossentry
  }
  \newcommand*{\setglossentrycompatibility}{%
    \let\glossentry\compatibleglossentry
    \let\subglossentry\compatiblesubglossentry
  }
  \setglossentrycompatibility
  \newcommand{\glossaryentryfield}[5]{%
    \GlossariesWarning
    {Deprecated use of \string\glossaryentryfield.^J
      I recommend you change to \string\glossentry.^J
      If you've just upgraded, try removing your gls auxiliary
      files^^J and recompile}%
    \noindent\textbf{\glstarget{#1}{#2}} #4 #3. #5\par}
  \newcommand*{\glossarysubentryfield}[6]{%
    \GlossariesWarning
    {Deprecated use of \string\glossarysubentryfield.^J
      I recommend you change to \string\subglossentry.^J
      If you've just upgraded, try removing your gls auxiliary
      files^^J and recompile}%

```

```

\glstarget{#2}{\strut}#4. #6\par}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newcommand*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\newcommand*{\@gls@getgrouptitle}[2]{%
  \dtl@ifsingle{#1}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
  }%
  {%
    \ifboolexpr{test{\ifstrequal{#1}{glssymbols}}
      or test{\ifstrequal{#1}{glsnumbers}}}%
    {%
      \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
    }%
    {%
      \def#2{#1}%
    }%
  }%
}
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%
  \DTLifint{#1}%
  {\edef#2{\char#1\relax}}%
  {%
    \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}}%
  }%
}
\newcommand*{\glsgetgrouplabel}[1]{%
\ifthenelse{\equal{#1}{\glssymbolsgroupname}}{\glssymbols}{%
\ifthenelse{\equal{#1}{\glsnumbersgroupname}}{\glsnumbers}{#1}}%
\newcommand*{\setentrycounter}[2][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1}%
  \fi
  \def\glsentrycounter{#2}%
}
\newcommand*{\setglossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \csname @glsstyle@#1\endcsname
  }%
}

```

```

\ifx\@glossary@default@style\relax
  \protected@edef\@glossary@default@style{#1}%
\fi
}
\newcommand*\@glossarystyle}[1]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \GlossariesWarning
    {Deprecated command \string\glossarystyle.^^J
      I recommend you switch to \string\setglossarystyle\space unless
      you want to maintain backward compatibility}%
    \setglossentrycompatibility
    \csname @glsstyle@#1\endcsname
    \ifcsdef{@glscompstyle@#1}%
    {\setglossentrycompatibility\cuse{@glscompstyle@#1}}%
    {}%
  }%
  \ifx\@glossary@default@style\relax
    \protected@edef\@glossary@default@style{#1}%
  \fi
}
\newcommand*\@newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand*\@renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
  }%
  {%
    \csdef{@glsstyle@#1}{#2}%
  }%
}
\newcommand*\@glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\@glshypernumber#1\nohyperpage{}}\@nil}
}

```

```

\def\glsnumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
    \else
      \@delimR#1\delimR\delimR\%
    \fi
    \ifx\#2\%
      \else
        #2%
      \fi
      \ifx\#3\%
        \else
          \glsnumber#3\@nil
        \fi
    }
\def\@delimR#1\delimR #2\delimR #3\{%
\ifx\#2\%
  \@delimN{#1}%
\else
  \gls@numberlink{#1}\delimR\gls@numberlink{#2}%
\fi}
\def\@delimN#1{\@delimN#1\delimN \delimN\}
\def\@delimN#1\delimN #2\delimN#3\{%
\ifx\#3\%
  \gls@numberlink{#1}%
\else
  \gls@numberlink{#1}\delimN\gls@numberlink{#2}%
\fi
}
\def\gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil
\endgroup}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
  \edef\@gls@x{\the\toks@}%
  \ifx\@gls@x\empty
    \else
      \hyperlink{\glsentrycounter\@gls@counterprefix\the\toks@}%
        {\the\toks@}%
    \fi
  \else
    \gls@ReturnAfterFi{%
      \gls@removespaces#2\@nil
    }%
  \fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glsnumber{#1}}}

```

```

\newcommand*{\hypersf}[1]{\textsf{\glshypernumber{#1}}}
\newcommand*{\hypertt}[1]{\texttt{\glshypernumber{#1}}}
\newcommand*{\hyperbf}[1]{\textbf{\glshypernumber{#1}}}
\newcommand*{\hypermd}[1]{\textmd{\glshypernumber{#1}}}
\newcommand*{\hyperit}[1]{\textit{\glshypernumber{#1}}}
\newcommand*{\hypersl}[1]{\textsl{\glshypernumber{#1}}}
\newcommand*{\hyperup}[1]{\textup{\glshypernumber{#1}}}
\newcommand*{\hypersc}[1]{\textsc{\glshypernumber{#1}}}
\newcommand*{\hyperemph}[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\gls@label]{%
  \def\gls@label{#2}%
  \newacronym[#4]{#1}{#2}{#3}%
  \ifcsundef{xspace}%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
    }%
  }%
  {%
    \expandafter\edef\csname#1\endcsname{%
      \noexpand\ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
        \noexpand\gls{#1}\noexpand\xspace}%
      }%
  }%
}
\newcommand{\newacronym}[4][{}]{
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{\short}
\newcommand*{\glsshortpluralkey}{\shortplural}
\newcommand*{\glslongkey}{\long}
\newcommand*{\glslongpluralkey}{\longplural}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}{%
    {\@acrfull{#1}{#2}[]}%
  }
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
\newrobustcmd{\glsspace}{\space}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\newcommand*\ns@Acrfull[2][{}]{%

```

```

\new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2} []}%
}
\def\@Acrfull#1#2[#3]{%
    \Acrfullfmt{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
    \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*{\ns@ACRfull[2] []}{%
    \new@ifnextchar[{\@ACRfull{#1}{#2}}%
        {\@ACRfull{#1}{#2} []}%
}
\def\@ACRfull#1#2[#3]{%
    \ACRfullfmt{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
    \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*{\ns@acrfullpl[2] []}{%
    \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
        {\@acrfullpl{#1}{#2} []}%
}
\def\@acrfullpl#1#2[#3]{%
    \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
    \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*{\ns@ACRfullpl[2] []}{%
    \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
        {\@ACRfullpl{#1}{#2} []}%
}
\def\@ACRfullpl#1#2[#3]{%
    \ACRfullplfmt{#1}{#2}{#3}%
}
}

```



```

\newcommand*{\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\gslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \gslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\gslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\gslongtok},%
      longplural={\the\gslongtok\noexpand\acrpluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink{##1}{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink{##1}{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink{##1}{##2}{%
    \mfirstucMakeUppercase{\genacrfullformat{##2}{##3}}}%
}

```

```

\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{\acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
  }%
}
\newcommand*{\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
\newcommand*{\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%

```

```

\PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
}%
}
\newcommand*\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
\ifglshaslong{glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glstrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*\acronymentry}[1]{\acronymfont{\glstryshort{##1}}}%
\renewcommand*\acronymsort}[2]{##1}%
\renewcommand*\acronymfont}[1]{##1}%
\renewcommand*\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*\acrpluralsuffix{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*\GenericAcronymFields{description={\the\glslongtok}}%
\renewcommand*\genacrfullformat}[2]{%
\glstrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}

```

```

}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \Glsentrylongpl{##1}##2\glsacspace{##1}%
  (\protect\firstacronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
  \settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\glsp{##1}(\acronymfont{\glentryshort{##1}})}%
}

```

```

}
\newacronymstyle{long-sc-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sc-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%

```

```

\GlsUseAcrStyleDefs{sm-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \mfirstucMakeUppercase
            {\glsentrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \mfirstucMakeUppercase
          {\glsentrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink{##1}{##2}{\glsentrylong{##2}##3\space

```

```

        (\acronymfont{\glentryshort{##2}})}%
\renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##2}})}%
\renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glentrylong{##2}##3\space
            (\acronymfont{\glentryshort{##2}})}%
\renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glentrylongpl{##2}##3\space
        (\acronymfont{\glentryshortpl{##2}})}%
\renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{%
        \mfirstucMakeUppercase{\glentrylongpl{##2}##3\space
            (\acronymfont{\glentryshortpl{##2}})}%
\renewcommand*{\glentryfull}[1]{%
    \glentrylong{##1}\space(\acronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\glentryfullpl}[1]{%
    \glentrylongpl{##1}\space(\acronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
    \GlsUseAcrEntryDisplayStyle{dua}%
}%
{%
    \GlsUseAcrStyleDefs{dua}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymentry}[1]{\acronymfont{\glentrylong{##1}}}%
    \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%

```



```

\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \firstacronymfont{\Glsentryshort{##1}}##2%
  \protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
  (\glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshort{##2}}##3\space
    (\glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
  (\glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \mfirstucMakeUppercase{\acronymfont{\glsentryshortpl{##2}}##3\space
    (\glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glsentrylongpl{##1})}%

```

```

}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote-sm}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
}

```

```

\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
  \let\acs\acrshort
  \let\Acs\Acrshort
  \let\acsp\acrshortpl
  \let\Acsp\Acrshortpl
  \let\acl\aclong
  \let\aclp\aclongpl
  \let\Acl\Aclong
  \let\Aclp\Aclongpl
  \let\acf\acrfull
  \let\acfp\acrfullpl
  \let\Acf\Acrfull
  \let\Acfp\Acrfullpl
  \let\ac\gls
  \let\Ac\Gls
  \let\acp\glspl
  \let\Acp\Glspl
}
\ifglsacrshortcuts
  \DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
  \defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\DefaultNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
      plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      firstplural={\acrfullformat{\noexpand\expandonce\noexpand\@glo@longpl}%
        {\noexpand\expandonce\noexpand\@glo@shortpl}},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@@gl@noexpand@assign@default@plurals
}
\newcommand*{\@@gl@noexpand@assign@default@plurals}{%
  \expandafter\ifx\csname glo@the\glslabeltok @plural\endcsname\@@gl@@default@acr@plural
    \csxdef{glo@the\glslabeltok @plural}{\expandonce\@glo@shortpl}%
  \fi
  \expandafter\ifx\csname glo@the\glslabeltok @firstpl\endcsname\@@gl@@default@acr@firstplural
    \csxdef{glo@the\glslabeltok @firstpl}{%
      \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)}%
  \fi
  \expandafter\ifx\csname glo@the\glslabeltok @descplural\endcsname\@@gl@@default@acr@descplural
    \csxdef{glo@the\glslabeltok @descplural}{\expandonce\@glo@longpl}%
  \fi
}
\newcommand*{\@@gl@@default@acr@plural}{\expandonce\@glo@shortpl}
\newcommand*{\@@gl@@default@acr@firstplural}{%
  \expandonce\@glo@longpl\glsspace(\expandonce\@glo@shortpl)%
}
\newcommand*{\@@gl@@default@acr@descplural}{\expandonce\@glo@longpl}
\newcommand*{\SetDefaultAcronymStyle}{%
  \@for\@gl@type:=\@gl@acronymlists\do{%
    \SetDefaultAcronymDisplayStyle{\@gl@type}%
  }%
}
\renewcommand{\newacronym}[4][{}]{%
  \ifx\@gl@acronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetDefaultAcronymDisplayStyle{\@glo@type}%
  \fi
  \gl@keylisttok{##1}%
  \gl@labeltok{##2}%
  \gl@shorttok{##3}%
  \gl@longtok{##4}%
  \newacronymhook

```

```

\DefaultNewAcronymDef
}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newcommand*{\acrfootnote}[3]{\acrlinkfootnote{#1}{#2}{#3}}
\newcommand*{\acrlinkfootnote}[3]{%
\footnote{\glslink{#1}{#2}{#3}}%
}
\newcommand*{\acrnolinkfootnote}[3]{%
\footnote{#3}%
}
\newcommand*{\SetDescriptionFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\acronymfont{\glsentryfmt}%
}%
{%
\firstacronymfont{\glsentryfmt}%
\ifglsymbol{\glslabel}%
{%
\expandafter\protect\expandafter\acrfootnote\expandafter
{\@gls@link@opts}{\@gls@link@label}%
{%
\glsifplural
{\glsentrysymbolplural{\glslabel}}%
{\glsentrysymbol{\glslabel}}%
}%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionFootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand\acronymfont{\the\glsshorttok}},%
sort={\the\glsshorttok},%
first={\the\glsshorttok},%
firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%

```

```

        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        symbol={\the\glslongtok},%
        symbolplural={\noexpand\expandonce\noexpand\@glo@longpl},%
        \the\glskeylisttok
    }%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
    \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionFootnoteAcronymStyle}{%
    \ifglsacrsmallcaps
        \@gls@deprecated@acr@warning{footnote,smallcaps,description}%
        {footnote-sc-desc}{short-sc-footnote-desc}%
    \else
        \ifglsacrsmaller
            \@gls@deprecated@acr@warning{footnote,smaller,description}%
            {footnote-sm-desc}{short-sm-footnote-desc}%
        \else
            \@gls@deprecated@acr@warning{footnote,description}%
            {footnote-desc}{short-footnote-desc}%
        \fi
    \fi
\renewcommand{\newacronym}[4][{}]{%
    \ifx\@glsacronymlists\@empty
        \def\@glo@type{\acronymtype}%
        \setkeys{glossentry}{##1}%
        \DeclareAcronymList{\@glo@type}%
        \SetDescriptionFootnoteAcronymDisplayStyle{\@glo@type}%
    \fi
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
    \DescriptionFootnoteNewAcronymDef
}%

```

```

\@for\@gls@type:=\@glsacronymlists\do{%
  \SetDescriptionFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
can’t both be set}{}%
\fi
}%
\newcommand*{\SetDescriptionDUAAcronymDisplayStyle}[1]{%
  \defglsentryfmt[##1]{\glsentryfmt}%
}
\newcommand*{\DescriptionDUANewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glslongtok},%
      sort={\the\glslongtok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@gls@expand@field{##1}{symbolplural}{##2}%
}

```

```

}%
\@do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionDUAAcronymStyle}{%
\ifglsacrsmallcaps
\PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
can’t both be set}{}%
\else
\ifglsacrsmaller
\PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
can’t both be set}{}%
\fi
\fi
\@gls@deprecated@acr@warning{dua,description}{dua-desc}{long-noshort-desc}%
\renewcommand{\newacronym}[4][]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionDUAAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionDUANewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionDUAAcronymDisplayStyle{\@gls@type}%
}%
}%
\newcommand*{\SetDescriptionAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\ifglsused{\glslabel}%
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty
\acronymfont{\glsgenentryfmt}\gls@org@insert
}%
}%
\glsgenentryfmt
\ifglshassymbol{\glslabel}%
{%
\glsifplural

```



```

        {%
        \def\@glo@symbol{\glsentrysymbolplural{\glslabel}}%
        }%
        {%
        \def\@glo@symbol{\glsentrysymbol{\glslabel}}%
        }%
        \space(\protect\firstacronymfont
        {\glscapscase
        {\@glo@symbol}
        {\@glo@symbol}
        {\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
    }%
    }%
    {\glscustomtext\glsinsert}%
}%
}
\newcommand*{\DescriptionNewAcronymDef}{%
\edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
sort={\the\glsshorttok},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
symbol={\noexpand\@glo@text},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\@gls@expand@field{##1}{symbolplural}{##2}%
}%

```

```

\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetDescriptionAcronymStyle}{%
\ifglsacrsmallcaps
\@gls@deprecated@acr@warning{smallcaps,description}%
{long-sc-short-desc}{long-short-sc-desc}%
\else
\ifglsacrsmaller
\@gls@deprecated@acr@warning{smaller,description}%
{long-sm-short-desc}{long-short-sm-desc}%
\else
\@gls@deprecated@acr@warning{description}%
{long-short-desc}{long-short-desc}%
\fi
\fi
\renewcommand{\newacronym}[4][\]{%
\ifx\@glsacronymlists\@empty
\def\@glo@type{\acronymtype}%
\setkeys{glossentry}{##1}%
\DeclareAcronymList{\@glo@type}%
\SetDescriptionAcronymDisplayStyle{\@glo@type}%
\fi
\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\DescriptionNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
\SetDescriptionAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
\renewcommand{\acronymfont}[1]{\textsc{##1}}
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
\else
\ifglsacrsmaller
\renewcommand*{\acronymfont}[1]{\textsmaller{##1}}%
\fi
\fi
}%
\newcommand*{\SetFootnoteAcronymDisplayStyle}[1]{%
\defglsentryfmt[#1]{%
\ifdefempty\glscustomtext
{%
\let\gls@org@insert\glsinsert
\let\glsinsert\@empty

```

```

\ifglused{\glslabel}%
{%
  \acronymfont{\glsgenentryfmt}\gls@org@insert
}%
{%
  \firstacronymfont{\glsgenentryfmt}\gls@org@insert
  \ifglshaslong{\glslabel}%
  {%
    \expandafter\protect\expandafter\acrfootnote\expandafter
    {\@gls@link@opts}{\@gls@link@label}%
    {%
      \glsifplural
        {\glsentrylongpl{\glslabel}}%
        {\glsentrylong{\glslabel}}%
    }%
  }%
}%
}%
}%
{\glscustomtext\glsinsert}%
}%
}
\newcommand*{\FootnoteNewAcronymDef}{%
\edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\noexpand\acronymfont{\the\glsshorttok}},%
    sort={\the\glsshorttok},%
    text={\the\glsshorttok},%
    plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    first={\the\glsshorttok},%
    firstplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
    description={\the\glslongtok},%
    descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
    \the\glskeylisttok
  }%
}%
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@descplural\gls@assign@descplural
\def\gls@assign@firstpl##1##2{%
  \@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@gls@expand@field{##1}{plural}{##2}%
}

```

```

}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@descplural\@org@gls@assign@descplural
}
\newcommand*\SetFootnoteAcronymStyle{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{footnote,smallcaps}{footnote-sc}{short-sc-footnote}%
  \else
    \ifglsacrsmaller
      \@gls@deprecated@acr@warning{footnote,smaller}{footnote-sm}{short-sm-footnote}%
    \else
      \@gls@deprecated@acr@warning{footnote}{footnote}{short-footnote}%
    \fi
  \fi
\renewcommand{\newacronym}[4][ ]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{\acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetFootnoteAcronymDisplayStyle{\@glo@type}%
  \fi
  \glskeylisttok{##1}%
  \glslabeltok{##2}%
  \glsshorttok{##3}%
  \glslongtok{##4}%
  \newacronymhook
  \FootnoteNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetFootnoteAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont[1]{\textsc{##1}}%
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \ifglsacrsmaller
    \renewcommand*\acronymfont[1]{\textsmaller{##1}}%
  \fi
\fi
\ifglsacrdua
  \PackageError{glossaries}{Option clash: ‘footnote’ and ‘dua’
    can’t both be set}{}%
\fi
}%
\DeclareRobustCommand*\glsdoparenifnotempty[2]{%

```

```

\protected@edef\gls@tmp{#1}%
\ifdefempty\gls@tmp
{}%
{%
  \ifx\gls@tmp\@gls@default@value
  \else
    \space (#2{#1})%
  \fi
}%
}
\newcommand*{\SetSmallAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#1]{%
    \ifdefempty\gls@customtext
    {%
      \let\gls@org@insert\glsinsert
      \let\glsinsert\@empty
      \ifglsused{\glslabel}%
      {%
        \acronymfont{\gls@genentryfmt}\gls@org@insert
      }%
      {%
        \gls@genentryfmt
        \ifgls@hasymbol{\glslabel}%
        {%
          \gls@ifplural
          {%
            \def\@glo@symbol{\gls@entrysymbolplural{\glslabel}}%
          }%
          {%
            \def\@glo@symbol{\gls@entrysymbol{\glslabel}}%
          }%
          \space
          (\gls@scapscase
          {\firstacronymfont{\@glo@symbol}}%
          {\firstacronymfont{\@glo@symbol}}%
          {\firstacronymfont{\mfirstucMakeUppercase{\@glo@symbol}}})%
        }%
        {}%
      }%
    }%
    {\gls@customtext\glsinsert}%
  }%
}
\newcommand*{\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\gls@shorttok}},%
      sort={\the\gls@shorttok},%
    }%
  }%
}

```

```

text={\the\glsshorttok},%
plural={\noexpand\expandonce\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
symbol={\the\glsshorttok},%
symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\SetSmallAcronymStyle}{%
  \ifglsacrsmallcaps
    \@gls@deprecated@acr@warning{smallcaps}{long-sc-short}{long-short-sc}%
  \else
    \@gls@deprecated@acr@warning{smaller}{long-sm-short}{long-short-sm}%
  \fi
\renewcommand{\newacronym}[4][\]{%
  \ifx\@glsacronymlists\@empty
    \def\@glo@type{acronymtype}%
    \setkeys{glossentry}{##1}%
    \DeclareAcronymList{\@glo@type}%
    \SetSmallAcronymDisplayStyle{\@glo@type}%
  \fi

```

```

\glskeylisttok{##1}%
\glslabeltok{##2}%
\glsshorttok{##3}%
\glslongtok{##4}%
\newacronymhook
\SmallNewAcronymDef
}%
\@for\@gls@type:=\@glsacronymlists\do{%
  \SetSmallAcronymDisplayStyle{\@gls@type}%
}%
\ifglsacrsmallcaps
  \renewcommand*\acronymfont}[1]{\textsc{##1}}
  \renewcommand*\acrpluralsuffix{\glsupacrpluralsuffix}%
\else
  \renewcommand*\acronymfont}[1]{\textsmaller{##1}}
\fi
\ifglsacrdua
  \ifglsacrsmallcaps
    \PackageError{glossaries}{Option clash: ‘smallcaps’ and ‘dua’
      can’t both be set}{}%
  \else
    \PackageError{glossaries}{Option clash: ‘smaller’ and ‘dua’
      can’t both be set}{}%
  \fi
\fi
}%
\newcommand*\SetDUADisplayStyle}[1]{%
  \defglsentryfmt[##1]{\glsentryfmt}%
}
\newcommand*\DUANewAcronymDef{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\the\glsshorttok},%
      text={\the\glslongtok},%
      first={\the\glslongtok},%
      plural={\noexpand\expandonce\noexpand\@glo@longpl},%
      firstplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\noexpand\expandonce\noexpand\@glo@longpl},%
      symbol={\the\glsshorttok},%
      symbolplural={\noexpand\expandonce\noexpand\@glo@shortpl},%
      \the\glskeylisttok
    }%
  }%
}%

```

```

\let\@org@gl@s@assign@firstpl\gl@s@assign@firstpl
\let\@org@gl@s@assign@plural\gl@s@assign@plural
\let\@org@gl@s@assign@symbolplural\gl@s@assign@symbolplural
\let\@org@gl@s@assign@descplural\gl@s@assign@descplural
\def\gl@s@assign@firstpl##1##2{%
  \@@gl@s@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@s@assign@plural##1##2{%
  \@@gl@s@expand@field{##1}{plural}{##2}%
}%
\def\gl@s@assign@symbolplural##1##2{%
  \@@gl@s@expand@field{##1}{symbolplural}{##2}%
}%
\def\gl@s@assign@descplural##1##2{%
  \@@gl@s@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@s@assign@firstpl\@org@gl@s@assign@firstpl
\let\gl@s@assign@plural\@org@gl@s@assign@plural
\let\gl@s@assign@symbolplural\@org@gl@s@assign@symbolplural
\let\gl@s@assign@descplural\@org@gl@s@assign@descplural
}
\newcommand*{\SetDUASStyle}{%
  \@gl@s@deprecated@acr@warning{dua}{dua}{long-noshort}%
  \renewcommand{\newacronym}[4][]{%
    \ifx\@gl@sacronymlists\@empty
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
      \SetDUADisplayStyle{\@glo@type}%
    \fi
    \gl@keylisttok{##1}%
    \gl@labeltok{##2}%
    \gl@shorttok{##3}%
    \gl@longtok{##4}%
    \newacronymhook
    \DUANewAcronymDef
  }%
  \@for\@gl@s@type:=\@gl@sacronymlists\do{%
    \SetDUADisplayStyle{\@gl@s@type}%
  }%
}
\newcommand*{\@gl@s@deprecated@acr@warning}[3]{%
  \GlossariesWarning{Acronym style option #1 deprecated.
    Use \string\setacronymstyle{#2} with just glossaries.sty or
    use \string\setabbreviationstyle[acronym]{#3} with
    glossaries-extra.sty}%
}
\newcommand*{\SetAcronymStyle}{%
  \SetDefaultAcronymStyle

```



```

\ifglsacrdescription
\ifglsacrfootnote
\SetDescriptionFootnoteAcronymStyle
\else
\ifglsacrdua
\SetDescriptionDUAAcronymStyle
\else
\SetDescriptionAcronymStyle
\fi
\fi
\else
\ifglsacrfootnote
\SetFootnoteAcronymStyle
\else
\ifthenelse{\boolean{glsacrsmallcaps}\OR
\boolean{glsacrsmaller}}{%
}%
\SetSmallAcronymStyle
}%
}%
\ifglsacrdua
\SetDUASyle
\fi
}%
\fi
\fi
}
\SetAcronymStyle
\newcommand*{\SetCustomDisplayStyle}[1]{%
\defglsentryfmt[#1]{\glsentryfmt}%
}
\newcommand*{\CustomAcronymFields}{%
name={\the\glsshorttok},%
description={\the\glslongtok},%
first={\acrfullformat{\the\glslongtok}{\the\glsshorttok}},%
firstplural={\acrfullformat
{\noexpand\glsentrylongpl{\the\glslabeltok}}%
{\noexpand\glsentryshortpl{\the\glslabeltok}}},%
text={\the\glsshorttok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix}%
}
\newcommand*{\CustomNewAcronymDef}{%
\protected@edef\@do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

        user1={\the\glsshorttok},%
        user2={\the\glsshorttok\noexpand\acrpluralsuffix},%
        user3={\the\glslongtok},%
        user4={\the\glslongtok\noexpand\acrpluralsuffix},%
        \CustomAcronymFields,%
        \the\glskeylisttok
    }%
}%
\@do@newglossaryentry
}
\newcommand*{\SetCustomStyle}{%
    \renewcommand{\newacronym}[4][{}]{%
        \ifx\@glsacronymlists\@empty
            \def\@glo@type{\acronymtype}%
            \setkeys{glossentry}{##1}%
            \DeclareAcronymList{\@glo@type}%
            \SetCustomDisplayStyle{\@glo@type}%
        \fi
        \glskeylisttok{##1}%
        \glslabeltok{##2}%
        \glsshorttok{##3}%
        \glslongtok{##4}%
        \newacronymhook
        \CustomNewAcronymDef
    }%
    \@for\@gls@type:=\@glsacronymlists\do{%
        \SetCustomDisplayStyle{\@gls@type}%
    }%
}
\RequirePackage{glossary-hypernav}[=v4.49]
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
    \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*{\showgloparent}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*{\showglolevel}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*{\showglotext}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*{\showgloplural}[1]{%
    \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}

```

```

\newcommand*{\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showgloftype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showgloname}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}
\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@short\endcsname
}

```

```

}
\newcommand*{\showglolong}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
  \expandafter\show\csname ifglo@\glsdetoklabel{#1}@flag\endcsname
}
\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \cshow{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*{\showglossaries}{%
  \show\@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glo@type@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glo@type@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glo@type@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glo@type@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glist@#1\endcsname
}
\csname ifglscompatible-2.07\endcsname
  \RequirePackage{glossaries-compatible-207}[v4.49]
\fi

```

8.19 Rollback v4.49 (glossaries-accsupp-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-accsupp}[2021/11/01 v4.49 (NLCT)
  Experimental glossaries accessibility]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\@ifpackageloaded{glossaries-extra}

```

```

{%
\ifx\@glsxtr@doaccsupp\empty
\GlossariesWarning{The 'glossaries-accsupp'
package has been loaded\MessageBreak
after the 'glossaries-extra' package. This\MessageBreak
can cause a failure to integrate both packages. \MessageBreak
Either use the 'accsupp' option when you load\MessageBreak
'glossaries-extra' or load 'glossaries-accsupp'\MessageBreak
before loading 'glossaries-extra'}%
\fi
}
{}
\def\compatibleglossentry#1#2{%
\toks@{#2}%
\protected@edef\@do@glossentry{%
\noexpand\accsuppglossaryentryfield{#1}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#1}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@glossentry
}
\def\compatiblesubglossentry#1#2#3{%
\toks@{#3}%
\protected@edef\@do@subglossentry{%
\noexpand\accsuppglossarysubentryfield{\number#1}%
{#2}%
{\noexpand\glsnamefont
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@name\endcsname}}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@desc\endcsname}%
{\expandafter\expandonce\csname glo@\glsdetoklabel{#2}@symbol\endcsname}%
{\the\toks@}%
}%
\@do@subglossentry
}
\RequirePackage{glossaries}[v4.49]
\providecommand{\gls@accsupp@engine}{accsupp}
\providecommand{\gls@accessibility}[4]{#4}
\ifdefstring\gls@accsupp@engine{accsupp}
{
\RequirePackage{accsupp}
\renewcommand{\gls@accessibility}[4]{%
\BeginAccSupp{#1,#2={#3}}#4\EndAccSupp{}}%
}
}
{}
\newcommand{\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}{useriiaaccess},%
  {user3access}{useriiiaaccess},%
  {user4access}{userivaccess},%
  {user5access}{uservaccess},%
  {user6access}{userviaccess}%
}
\def\@gls@noaccess{\relax}
\let\@gls@oldnewglossaryentryprehook\@newglossaryentryprehook
\renewcommand*{\@newglossaryentryprehook}{%
  \@gls@oldnewglossaryentryprehook
  \def\@glo@access{\relax}%
  \def\@glo@textaccess{\@glo@access}%
  \def\@glo@firstaccess{\@glo@access}%
  \def\@glo@pluralaccess{\@glo@textaccess}%
  \def\@glo@firstpluralaccess{\@glo@pluralaccess}%
  \def\@glo@symbolaccess{\relax}%
  \def\@glo@symbolpluralaccess{\@glo@symbolaccess}%
  \def\@glo@descaccess{\relax}%
  \def\@glo@descpluralaccess{\@glo@descaccess}%
  \def\@glo@shortaccess{\relax}%
  \def\@glo@shortpluralaccess{\@glo@shortaccess}%
  \def\@glo@longaccess{\relax}%
  \def\@glo@longpluralaccess{\@glo@longaccess}%
  \def\@glo@useriaccess{\relax}%
  \def\@glo@useriiaaccess{\relax}%
  \def\@glo@useriiiaaccess{\relax}%
}

```

```

\def\@glo@userivaccess{\relax}%
\def\@glo@uservaccess{\relax}%
\def\@glo@userviaccess{\relax}%
}
\let\@gls@oldnewglossaryentryposthook\@newglossaryentryposthook
\renewcommand*{\@newglossaryentryposthook}{%
\@gls@oldnewglossaryentryposthook
\expandafter
\protected@xdef\csname glo@\@glo@label @access\endcsname{%
\@glo@access}%
\expandafter
\protected@xdef\csname glo@\@glo@label @textaccess\endcsname{%
\@glo@textaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstaccess\endcsname{%
\@glo@firstaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @pluralaccess\endcsname{%
\@glo@pluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @firstpluralaccess\endcsname{%
\@glo@firstpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolaccess\endcsname{%
\@glo@symbolaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @symbolpluralaccess\endcsname{%
\@glo@symbolpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descaccess\endcsname{%
\@glo@descaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @descpluralaccess\endcsname{%
\@glo@descpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortaccess\endcsname{%
\@glo@shortaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @shortpluralaccess\endcsname{%
\@glo@shortpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longaccess\endcsname{%
\@glo@longaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @longpluralaccess\endcsname{%
\@glo@longpluralaccess}%
\expandafter
\protected@xdef\csname glo@\@glo@label @useriaccess\endcsname{%
\@glo@useriaccess}%
\expandafter

```



```

\protected@xdef\csname glo@\glo@label @useriiaccess\endcsname{%
  \@glo@useriiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @useriiiaccess\endcsname{%
  \@glo@useriiiaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userivaccess\endcsname{%
  \@glo@userivaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @uservaccess\endcsname{%
  \@glo@uservaccess}%
\expandafter
\protected@xdef\csname glo@\glo@label @userviaccess\endcsname{%
  \@glo@userviaccess}%
}
\newcommand*{\glsentryaccess}[1]{%
  \@gls@entry@field{#1}{access}%
}
\newcommand*{\glsentrytextaccess}[1]{%
  \@gls@entry@field{#1}{textaccess}%
}
\newcommand*{\glsentryfirstaccess}[1]{%
  \@gls@entry@field{#1}{firstaccess}%
}
\newcommand*{\glsentrypluralaccess}[1]{%
  \@gls@entry@field{#1}{pluralaccess}%
}
\newcommand*{\glsentryfirstpluralaccess}[1]{%
  \@gls@entry@field{#1}{firstpluralaccess}%
}
\newcommand*{\glsentrysymbolaccess}[1]{%
  \@gls@entry@field{#1}{symbolaccess}%
}
\newcommand*{\glsentrysymbolpluralaccess}[1]{%
  \@gls@entry@field{#1}{symbolpluralaccess}%
}
\newcommand*{\glsentrydescaccess}[1]{%
  \@gls@entry@field{#1}{descaccess}%
}
\newcommand*{\glsentrydescpluralaccess}[1]{%
  \@gls@entry@field{#1}{descpluralaccess}%
}
\newcommand*{\glsentryshortaccess}[1]{%
  \@gls@entry@field{#1}{shortaccess}%
}
\newcommand*{\glsentryshortpluralaccess}[1]{%
  \@gls@entry@field{#1}{shortpluralaccess}%
}
\newcommand*{\glsentrylongaccess}[1]{%
  \@gls@entry@field{#1}{longaccess}%
}

```

```

}
\newcommand*{\glsentrylongpluralaccess}[1]{%
  \@gls@entry@field{#1}{longpluralaccess}%
}
\newcommand*{\glsentryuseriaccess}[1]{%
  \@gls@entry@field{#1}{useriaccess}%
}
\newcommand*{\glsentryuseriiaccess}[1]{%
  \@gls@entry@field{#1}{useriiaccess}%
}
\newcommand*{\glsentryuseriiiaccess}[1]{%
  \@gls@entry@field{#1}{useriiiaccess}%
}
\newcommand*{\glsentryuserivaccess}[1]{%
  \@gls@entry@field{#1}{userivaccess}%
}
\newcommand*{\glsentryuservaccess}[1]{%
  \@gls@entry@field{#1}{uservaccess}%
}
\newcommand*{\glsentryuserviaccess}[1]{%
  \@gls@entry@field{#1}{userviaccess}%
}
\newcommand{\glsfieldaccsupp}[4]{%
  \ifdef\glscategory
  {%
    \ifcsdef{glstr\glscategory{#4}#3accsupp}%
    {\csname glstr\glscategory{#4}#3accsupp\endcsname{#1}{#2}}%
    {%
      \ifcsdef{glstr\glscategory{#4}accsupp}%
      {\csname glstr\glscategory{#4}accsupp\endcsname{#1}{#2}}%
      {%
        \ifcsdef{gls#3accsupp}%
        {\csname gls#3accsupp\endcsname{#1}{#2}}%
        {\glsaccsupp{#1}{#2}}%
      }%
    }%
  }%
  {%
    \ifcsdef{gls#3accsupp}%
    {\csname gls#3accsupp\endcsname{#1}{#2}}%
    {\glsaccsupp{#1}{#2}}%
  }%
}
\newcommand{\xglsfieldaccsupp}[1]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsfieldaccsupp\expandafter{\@gls@replacementtext}%
}
\newcommand*{\glsshortaccsupp}[2]{\glsaccessibility{E}{#1}{#2}}
\newcommand*{\glsshortplaccsupp}{\glsshortaccsupp}
\newcommand*{\glsaccsupp}[2]{\glsaccessibility{ActualText}{#1}{#2}}

```

```

\newcommand*{\xglsaccsupp}[2]{%
  \protected@edef\@gls@replacementtext{#1}%
  \expandafter\glsaccsupp\expandafter{\@gls@replacementtext}{#2}%
}
\newcommand*{\@gls@access@display}[2]{%
  \protected@edef\@glo@access{#2}%
  \ifx\@glo@access\@gls@noaccess
    #1%
  \else
    \xglsaccsupp{\@glo@access}{#1}%
  \fi
}
\newcommand*{\@gls@fieldaccess@display}[4]{%
  \protected@edef\@glo@access{#4}%
  \ifdefequal\@glo@access\@gls@noaccess
    {#3}%
  {\expandafter\glsfieldaccsupp\expandafter{\@glo@access}{#3}{#2}{#1}}%
}
\newrobustcmd*{\glsnameaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@access}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{name}{#1}{\glsentryaccess{#2}}}%
}
\newrobustcmd*{\glstextaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@textaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{text}{#1}{\glsentrytextaccess{#2}}}%
}
\newrobustcmd*{\glspluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@pluralaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{plural}{#1}{\glsentrypluralaccess{#2}}}%
}
\newrobustcmd*{\glsfirstaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{first}{#1}{\glsentryfirstaccess{#2}}}%
}
\newrobustcmd*{\glsfirstpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@firstpluralaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{firstpl}{#1}{\glsentryfirstpluralaccess{#2}}}%
}
\newrobustcmd*{\glssymbolaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolaccess}%
    {#1}%
  {\@gls@fieldaccess@display{#2}{symbol}{#1}{\glsentrysymbolaccess{#2}}}%
}
\newrobustcmd*{\glssymbolpluralaccessdisplay}[2]{%
  \ifcsundef{glo@\glsdetoklabel{#2}@symbolpluralaccess}%

```

```

{#1}%
{\@gls@fieldaccess@display{#2}{symbolplural}{#1}{\glsentrysymbolpluralaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@descaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{desc}{#1}{\glsentrydescaccess{#2}}}%
}
\newrobustcmd*{\glsdescriptionpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@descpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{descplural}{#1}{\glsentrydescpluralaccess{#2}}}%
}
\newrobustcmd*{\glsshortaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@shortaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{short}{#1}{\glsentryshortaccess{#2}}}%
}
\newrobustcmd*{\glsshortpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@shortpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{shortpl}{#1}{\glsentryshortpluralaccess{#2}}}%
}
\newrobustcmd*{\glslongaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@longaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{long}{#1}{\glsentrylongaccess{#2}}}%
}
\newrobustcmd*{\glslongpluralaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@longpluralaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{longpl}{#1}{\glsentrylongpluralaccess{#2}}}%
}
\newrobustcmd*{\glsuseriaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useri}{#1}{\glsentryuseriaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userii}{#1}{\glsentryuseriiaccess{#2}}}%
}
\newrobustcmd*{\glsuseriiiaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@useriiiaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{useriii}{#1}{\glsentryuseriiiaccess{#2}}}%
}
\newrobustcmd*{\glsuserivaccessdisplay}[2]{%
\ifcsundef{glo@glstetoklabel{#2}@userivaccess}%

```

```

{#1}%
{\@gls@fieldaccess@display{#2}{useriv}{#1}{\glsentryuserivaccess{#2}}}%
}
\newrobustcmd*{\glsuservaccessdisplay}[2]{%
\ifcsundef{gls@glstoklabel{#2}@uservaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{userv}{#1}{\glsentryuservaccess{#2}}}%
}
\newrobustcmd*{\glsuserviaccessdisplay}[2]{%
\ifcsundef{gls@glstoklabel{#2}@userviaccess}%
{#1}%
{\@gls@fieldaccess@display{#2}{uservi}{#1}{\glsentryuserviaccess{#2}}}%
}
\newrobustcmd*{\glsaccessdisplay}[3]{%
\ifcsundef{gls#1accessdisplay}%
{%
\PackageError{glossaries-accsupp}{No accessibility support
for key ‘#1’}{}%
}%
{%
\csname gls#1accessdisplay\endcsname{#2}{#3}%
}%
}
\renewcommand*{\@gls@default@entryfmt}[2]{%
\ifdefempty{glscustomtext}
{%
\glsifplural
{%
\glscapscase
{%
\ifglsused{glslabel}
{%
#2{\glspluralaccessdisplay
{\glsentryplural{glslabel}}{glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{glslabel}}{glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{glslabel}}{glslabel}}
{\glsinsert}%
}%
}%
#1{\glsfirstpluralaccessdisplay
{\glsentryfirstplural{glslabel}}{glslabel}}%
{\glsdescriptionpluralaccessdisplay
{\glsentrydescplural{glslabel}}{glslabel}}%
{\glsymbolpluralaccessdisplay
{\glsentrysymbolplural{glslabel}}{glslabel}}%
{\glsinsert}%
}%
}%
}

```

```

{%
  \ifglused\glslabel
  {%
    #2{\glspluralaccessdisplay
      {\Glsentryplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glsfirstpluralaccessdisplay
      {\Glsentryfirstplural{\glslabel}}{\glslabel}}%
    {\glsdescriptionpluralaccessdisplay
      {\glsentrydescplural{\glslabel}}{\glslabel}}%
    {\glssymbolpluralaccessdisplay
      {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
{%
  \ifglused\glslabel
  {%
    \MakeUppercase{%
      #2{\glspluralaccessdisplay
        {\glsentryplural{\glslabel}}{\glslabel}}%
        {\glsdescriptionpluralaccessdisplay
          {\glsentrydescplural{\glslabel}}{\glslabel}}%
        {\glssymbolpluralaccessdisplay
          {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
        {\glsinsert}}}%
    }%
    {%
      \MakeUppercase{%
        #1{\glsfirstpluralaccessdisplay
          {\glsentryfirstplural{\glslabel}}{\glslabel}}%
          {\glsdescriptionpluralaccessdisplay
            {\glsentrydescplural{\glslabel}}{\glslabel}}%
          {\glssymbolpluralaccessdisplay
            {\glsentrysymbolplural{\glslabel}}{\glslabel}}%
          {\glsinsert}}}%
        }%
      }%
    }%
  }%
  \glscapscase
  {%
    \ifglused\glslabel
    {%

```

```

#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
{%
#1{\glsfirstaccessdisplay
  {\glentryfirst{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%
}%
}%
{%
\ifglsused\glslabel
{%
\MakeUppercase{%
#2{\glstextaccessdisplay
  {\glentrytext{\glslabel}}{\glslabel}}%
  {\glsdescriptionaccessdisplay
    {\glentrydesc{\glslabel}}{\glslabel}}%
  {\glssymbolaccessdisplay
    {\glentrysymbol{\glslabel}}{\glslabel}}%
  {\glsinsert}}%

```

```

}%
{%
  \MakeUppercase{%
    #1{\glsfirstaccessdisplay
      {\glsentryfirst{\glslabel}}{\glslabel}}%
      {\glsdescriptionaccessdisplay
        {\glsentrydesc{\glslabel}}{\glslabel}}%
      {\glssymbolaccessdisplay
        {\glsentrysymbol{\glslabel}}{\glslabel}}%
      {\glsinsert}}%
  }%
}%
}%
}%
{%
  \ifglsused{\glslabel}%
  {%
    #2{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glssymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
  {%
    #1{\glscustomtext}%
    {\glsdescriptionaccessdisplay
      {\glsentrydesc{\glslabel}}{\glslabel}}%
    {\glssymbolaccessdisplay
      {\glsentrysymbol{\glslabel}}{\glslabel}}%
    {\glsinsert}}%
  }%
}%
}
\renewcommand*{\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}%
}%
{%
\glsfirstpluralacessdisplay
{\mfirstucMakeUppercase{\glsentryfirstplural{\glslabel}}}%
{\glslabel}%
\mfirstucMakeUppercase{\glsinsert}%
}%
}%
}%
{%
\glscapscase
{%
\ifglsused\glslabel
{%
\glstextaccessdisplay{\glsentrytext{\glslabel}}{\glslabel}%
\glsinsert
}%
{%
\glsfirstaccessdisplay{\glsentryfirst{\glslabel}}{\glslabel}%
\glsinsert
}%
}%
{%
\ifglsused\glslabel
{%
\glstextaccessdisplay{\Glsentrytext{\glslabel}}{\glslabel}%
\glsinsert

```

```

    }%
    {%
      \glsfirstaccessdisplay{\Glsentryfirst{\glslabel}}{\glslabel}%
      \glsinsert
    }%
  }%
  {%
    \ifglssused\glslabel
    {%
      \glstextaccessdisplay
      {\mfirstucMakeUppercase{\glsentrytext{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
    {%
      \glsfirstaccessdisplay
      {\mfirstucMakeUppercase{\glsentryfirst{\glslabel}}}{\glslabel}%
      \mfirstucMakeUppercase{\glsinsert}%
    }%
  }%
}
}
}
}
{%
  \glscustomtext\glsinsert
}%
}
\renewcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglssused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \acronymfont
          {\glsshortpluralaccessdisplay
            {\Glsentryshortpl{\glslabel}}{\glslabel}}%
          \glsinsert
        }%
        {%
          \mfirstucMakeUppercase
          {\acronymfont
            {\glsshortpluralaccessdisplay
              {\glsentryshortpl{\glslabel}}{\glslabel}}%

```

```

        \glsinsert}%
    }%
}%
{%
    \glscapscase
    {%
        \acronymfont
        {\glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \acronymfont
        {\glsshortaccessdisplay{\Glsentryshort{\glslabel}}{\glslabel}}%
        \glsinsert
    }%
    {%
        \mfirstucMakeUppercase
        {\acronymfont{%
            \glsshortaccessdisplay{\glsentryshort{\glslabel}}{\glslabel}}%
            \glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \mfirstucMakeUppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
{%
    \glscapscase
    {%
        \genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \Genacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
        \mfirstucMakeUppercase
        {\genacrfullformat{\glslabel}{\glsinsert}}%
    }%
}%

```

```

    }%
  }%
}%
{%
  \glscustomtext
}%
}
\renewcommand*{\genacrfullformat}[2]{%
  \glslongaccessdisplay{\glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\glsentryshort{#1}}}{#1})%
}
\renewcommand*{\Genacrfullformat}[2]{%
  \glslongaccessdisplay{\Glsentrylong{#1}}{#1}#2\space
  (\glsshortaccessdisplay{\protect\firstacronymfont{\Glsentryshort{#1}}}{#1})%
}
\renewcommand*{\genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glsentryshortpl{#1}}}{#1})%
}
\renewcommand*{\Genplacrfullformat}[2]{%
  \glslongpluralaccessdisplay{\Glsentrylongpl{#1}}{#1}#2\space
  (\glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glsentryshortpl{#1}}}{#1})%
}
\def\@acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\def\@Acrshort#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \let\glsifplural\@secondoftwo
    \let\glsapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glsshortaccessdisplay{\Glsentryshort{#2}}{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%

```

```

}%
\glspostlinkhook
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glsshortaccessdisplay
{\MakeUppercase{\glentryshort{#2}}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\glentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@Acrlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glslongaccessdisplay{\Glsentrylong{#2}}{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@ACRlong#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\relax

```

```

\let\glsifplural\@secondoftwo
\let\glsupcase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
  \acronymfont{\glslongaccessdisplay{%
    \MakeUppercase{\glsentrylong{#2}}}{#2}#3}%
  }%
  \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\def\@gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\glsentrytext{#2}}}{#2}#3}%
}
\def\@Gls@text@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@text@accessdisplay{\Glsentrytext{#2}}}{#2}#3}%
}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@text@accessdisplay{\mfirstucMakeUppercase{\glsentrytext{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\glsentryfirst{#2}}}{#2}#3}%
}
\def\@Gls@first@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@first@accessdisplay{\Glsentryfirst{#2}}}{#2}#3}%
}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@first@accessdisplay{\mfirstucMakeUppercase{\glsentryfirst{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\glsentryplural{#2}}}{#2}#3}%
}
\def\@Gls@plural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@plural@accessdisplay{\Glsentryplural{#2}}}{#2}#3}%
}
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{%
    {\gls@plural@accessdisplay{\mfirstucMakeUppercase{\glsentryplural{#2}}}{#2}%
    \mfirstucMakeUppercase{#3}}}%
}
\def\@gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\glsentryfirstplural{#2}}}{#2}#3}%
}
\def\@Gls@firstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\gls@firstplural@accessdisplay{\Glsentryfirstplural{#2}}}{#2}#3}%
}

```

```

\def\@GLSfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsfirstpluralaccessdisplay{\mfirstucMakeUppercase{\glsentryfirstplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\glsentryname{#2}}{#2}#3}%
}
\def\@Glsname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsnameaccessdisplay{\Glsentryname{#2}}{#2}#3}%
}
\def\@GLSname@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsnameaccessdisplay{\mfirstucMakeUppercase{\glsentryname{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\glsentrydesc{#2}}{#2}#3}%
}
\def\@Glsdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionaccessdisplay{\Glsentrydesc{#2}}{#2}#3}%
}
\def\@GLSdesc@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionaccessdisplay{\mfirstucMakeUppercase{\glsentrydesc{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\glsentrydescplural{#2}}{#2}#3}%
}
\def\@Glsdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsdescriptionpluralaccessdisplay{\Glsentrydescplural{#2}}{#2}#3}%
}
\def\@GLSdescplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsdescriptionpluralaccessdisplay{\mfirstucMakeUppercase{\glsentrydescplural{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\glsentrysymbol{#2}}{#2}#3}%
}
\def\@Glsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glssymbolaccessdisplay{\Glsentrysymbol{#2}}{#2}#3}%
}
\def\@GLSsymbol@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glssymbolaccessdisplay{\mfirstucMakeUppercase{\glsentrysymbol{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glssymbolplural@#1#2[#3]{%

```

```

\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glentrysymbolplural{#2}}{#2}#3}%
}
\def\@Glsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glssymbolpluralaccessdisplay{\glentrysymbolplural{#2}}{#2}#3}%
}
\def\@GLSsymbolplural@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glssymbolpluralaccessdisplay{\mfirstucMakeUppercase{\glentrysymbolplural{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\glentryuseri{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriaccessdisplay{\Glentryuseri{#2}}{#2}#3}%
}
\def\@GLSuseri@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriaccessdisplay{\mfirstucMakeUppercase{\glentryuseri{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\glentryuserii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiaccessdisplay{\Glentryuserii{#2}}{#2}#3}%
}
\def\@GLSuserii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriiaccessdisplay{\mfirstucMakeUppercase{\glentryuserii{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuseriii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\glentryuseriii{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuseriiiaccessdisplay{\Glentryuseriii{#2}}{#2}#3}%
}
\def\@GLSuseriii@#1#2[#3]{%
\@gls@field@link{#1}{#2}{%
\glsuseriiiaccessdisplay{\mfirstucMakeUppercase{\glentryuseriii{#2}}}{#2}%
\mfirstucMakeUppercase{#3}}%
}
\def\@glsuseriv@#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\glentryuseriv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
\@gls@field@link{#1}{#2}{\glsuserivaccessdisplay{\Glentryuseriv{#2}}{#2}#3}%
}
\def\@GLSuseriv@#1#2[#3]{%

```



```

\@gls@field@link{#1}{#2}%
  {\glsuserivaccessdisplay{\mfirstucMakeUppercase{\glsentryuseriv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\glsentryuserv{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuservaccessdisplay{\Glsentryuserv{#2}}{#2}#3}%
}
\def\@GLSuserv@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuservaccessdisplay{\mfirstucMakeUppercase{\glsentryuserv{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\def\@glsuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\glsentryuservi{#2}}{#2}#3}%
}
\def\@Glsuser@i#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsuserviaccessdisplay{\Glsentryuservi{#2}}{#2}#3}%
}
\def\@GLSuservi@#1#2[#3]{%
  \@gls@field@link{#1}{#2}%
  {\glsuserviaccessdisplay{\mfirstucMakeUppercase{\glsentryuservi{#2}}}{#2}%
   \mfirstucMakeUppercase{#3}}%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentryname}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsnamefont{\glsnameaccessdisplay{\Glsentryname{#1}}{#1}}%
  }%
}
\renewcommand*{\glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\glsentrydesc{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrydesc}[1]{%
  \glsdoifexists{#1}%
  {%
    \glsdescriptionaccessdisplay{\Glsentrydesc{#1}}{#1}%
  }%
}

```

```

\renewcommand*{\glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\glsentrysymbol{#1}}{#1}%
  }%
}
\renewcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexists{#1}%
  {%
    \glssymbolaccessdisplay{\Glsentrysymbol{#1}}{#1}%
  }%
}
\newcommand*{\accsuppglossaryentryfield}[5]{%
  \glossaryentryfield{#1}%
  {\glsnameaccessdisplay{#2}{#1}}%
  {\glsdescriptionaccessdisplay{#3}{#1}}%
  {\glssymbolaccessdisplay{#4}{#1}}{#5}%
}
\newcommand*{\accsuppglossarysubentryfield}[6]{%
  \glossarysubentryfield{#1}{#2}%
  {\glsnameaccessdisplay{#3}{#2}}%
  {\glsdescriptionaccessdisplay{#4}{#2}}%
  {\glssymbolaccessdisplay{#5}{#2}}{#6}%
}
\renewacronymstyle{long-short}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \glslongaccessdisplay{\Glsentrylong{##1}}{##1}##2\space
    (\glsshortaccessdisplay
      {\protect\firstacronymfont{\glsentryshort{##1}}}{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}##2\space
    (\glsshortpluralaccessdisplay
      {\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1})%
  }%
}

```

```

\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glpluralsuffix}%
}
\renewacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
  \glsshortaccessdisplay
    {\protect\firstacronymfont{\Glentryshort{##1}}}{##1}##2\space
  (\glslongaccessdisplay{\glentrylong{##1}}{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \glsshortpluralaccessdisplay
    {\protect\firstacronymfont{\Glentryshortpl{##1}}}{##1}##2\space
  (\glslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glpluralsuffix}%
}
\renewacronymstyle{long-short-desc}%
{%
  \GlsUseAcrEntryDispStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%

```

```

\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{long-sm-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sm-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sm-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}
\renewacronymstyle{sc-short-long-desc}%
{%
\GlsUseAcrEntryDispStyle{sc-short-long}%
}%
{%
\GlsUseAcrStyleDefs{sc-short-long}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
(\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}))%
}

```

```

}
\renewacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDispStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
}
\renewacronymstyle{dua}%
{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glslongpluralaccessdisplay{\glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay{\Glsentrylongpl{\glslabel}}{\glslabel}%
          \glsinsert
        }%
        {%
          \glslongpluralaccessdisplay
            {\mfirstucMakeUppercase{\glsentrylongpl{\glslabel}}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
      {%
        \glscapscase
        {%
          \glslongaccessdisplay{\glsentrylong{\glslabel}}{\glslabel}\glsinsert
        }%
        {%
          \glslongaccessdisplay{\Glsentrylong{\glslabel}}{\glslabel}\glsinsert
        }%
        {%
          \glslongaccessdisplay
            {\mfirstucMakeUppercase
              {\glsentrylong{\glslabel}\glsinsert}}{\glslabel}%
            {\mfirstucMakeUppercase{\glsinsert}}%
        }%
      }%
    }%
  }%
}

```

```

    }%
  }%
  {%
    \glsgenentryfmt
  }%
}%
{\glscustomtext\glsinsert}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay{\Glsentrylong{##2}}{##2}##3\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongaccessdisplay
      {\mfirstucMakeUppercase{\glsentrylong{##2}}{##2}##3\space
      (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##2}}}{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\Glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glslongpluralaccessdisplay
      {\mfirstucMakeUppercase{\glsentrylongpl{##2}}{##2}##3\space
      (\glsshortpluralaccessdisplay
        {\acronymfont{\glsentryshortpl{##2}}}{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \glslongaccessdisplay{\glsentrylong{##1}}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfull}[1]{%
  \glslongaccessdisplay{\Glsentrylong{##1}}{##1}\space
  (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})%
}%
\renewcommand*{\glsentryfullpl}[1]{%

```

```

\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\glslongpluralaccessdisplay{\Glsentrylongpl{##1}}{##1}\space
(\glsshortpluralaccessdisplay{\acronymfont{\glsentryshortpl{##1}}}{##1})%
}%
\renewcommand*{\acronymentry}[1]{%
\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\renewacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{%
\glslongaccessdisplay{\acronymfont{\glsentrylong{##1}}}{##1}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\renewacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\glsshortaccessdisplay
{\protect\firstacronymfont{\glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\glsshortaccessdisplay
{\firstacronymfont{\Glsentryshort{##1}}}{##1}##2%
\protect\footnote{\glslongaccessdisplay{\glsentrylong{##1}}{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\glsshortpluralaccessdisplay
{\protect\firstacronymfont{\Glsentryshortpl{##1}}}{##1}##2%
\protect\footnote{\glslongpluralaccessdisplay{\glsentrylongpl{##1}}{##1}}%
}

```

```

}%
\renewcommand*{\acronymentry}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##2}}}{##2}##3\space
    (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshort{##2}}}{##2}##3\space
        (\gslongaccessdisplay{\glentrylong{##2}}{##2}))}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
      (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsshortpluralaccessdisplay
      {\mfirstucMakeUppercase
        {\acronymfont{\glentryshortpl{##2}}}{##2}##3\space
        (\gslongpluralaccessdisplay{\glentrylongpl{##2}}{##2}))}}%
\renewcommand*{\glentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\gslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}\space
  (\gslongaccessdisplay{\glentrylong{##1}}{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\gslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \glsshortpluralaccessdisplay
    {\acronymfont{\glentryshortpl{##1}}}{##1}\space
    (\gslongpluralaccessdisplay{\glentrylongpl{##1}}{##1})}%

```



```

}
\renewacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glstextup{\glspluralsuffix}}%
}%
\renewacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{%
    \glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}%
\renewacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sc-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sc}%
}%
{%
  \GlsUseAcrStyleDefs{footnote-sc}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glentryshort{##1}}}{##1})}%
}
\renewacronymstyle{footnote-sm-desc}%
{%
  \GlsUseAcrEntryDispStyle{footnote-sm}%
}

```

```

}%
{%
  \GlsUseAcrStyleDefs{footnote-sm}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glslongaccessdisplay{\glsentrylong{##1}}{##1}\space
    (\glsshortaccessdisplay{\acronymfont{\glsentryshort{##1}}}{##1})}%
  }
  \newcommand*{\glsdefaultshortaccess}[2]{#1}
  \renewcommand*{\newacronymhook}{%
    \protected@edef\@gls@keylist{%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      shortpluralaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      \the\glskeylisttok}%
    \expandafter\glskeylisttok\expandafter{\@gls@keylist}%
  }
  \renewcommand*{\DefaultNewAcronymDef}{%
    \edef\@do@newglossaryentry{%
      \noexpand\newglossaryentry{\the\glslabeltok}%
      {%
        type=\acronymtype,%
        name={\the\glsshorttok},%
        description={\the\glslongtok},%
        descriptionaccess=\relax,%
        text={\the\glsshorttok},%
        access={\noexpand\@glo@textaccess},%
        sort={\the\glsshorttok},%
        short={\the\glsshorttok},%
        shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
        shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
        long={\the\glslongtok},%
        longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
        first={\noexpand\glslongaccessdisplay
          {\the\glslongtok}{\the\glslabeltok}\space
          (\noexpand\glsshortaccessdisplay
            {\the\glsshorttok}{\the\glslabeltok})},%
        plural={\the\glsshorttok\acrpluralsuffix},%
        firstplural={\noexpand\glslongpluralaccessdisplay
          {\noexpand\@glo@longpl}{\the\glslabeltok}\space
          (\noexpand\glsshortpluralaccessdisplay
            {\noexpand\@glo@shortpl}{\the\glslabeltok})},%
        firstaccess=\relax,%
        firstpluralaccess=\relax,%
        textaccess={\noexpand\@glo@shortaccess},%
        \the\glskeylisttok
      }%
    }%
  }
  \let\@org@gls@assign@firstpl\gls@assign@firstpl

```

```

\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@descplural\gl@assign@descplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@descplural##1##2{%
  \@gl@expand@field{##1}{descplural}{##2}%
}%
\do@newglossaryentry
\let\gl@assign@firstpl\@org@gl@assign@firstpl
\let\gl@assign@plural\@org@gl@assign@plural
\let\gl@assign@symbolplural\@org@gl@assign@symbolplural
\@gl@noexpand@assign@default@plurals
}
\renewcommand*{\DescriptionFootnoteNewAcronymDef}{%
  \edef\do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\gl@labeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\gl@shorttok}},%
      sort={\the\gl@shorttok},%
      text={\the\gl@shorttok},%
      short={\the\gl@shorttok},%
      shortplural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
      shortaccess={\gl@defaultshortaccess{\the\gl@longtok}{\the\gl@shorttok}},%
      long={\the\gl@longtok},%
      longplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\gl@shorttok\noexpand\acrpluralsuffix},%
      symbol={\the\gl@longtok},%
      symbolplural={\the\gl@longtok\noexpand\acrpluralsuffix},%
      firstpluralaccess=\relax,
      textaccess={\noexpand\@glo@shortaccess},%
      \the\gl@keylisttok
    }%
  }%
\let\@org@gl@assign@firstpl\gl@assign@firstpl
\let\@org@gl@assign@plural\gl@assign@plural
\let\@org@gl@assign@symbolplural\gl@assign@symbolplural
\def\gl@assign@firstpl##1##2{%
  \@gl@expand@field{##1}{firstpl}{##2}%
}%
\def\gl@assign@plural##1##2{%
  \@gl@expand@field{##1}{plural}{##2}%
}%
\def\gl@assign@symbolplural##1##2{%
  \@gl@expand@field{##1}{symbolplural}{##2}%
}%

```

```

}%
\do@newglossaryentry
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@symbolplural\org@gls@assign@symbolplural
}
\renewcommand*{\DescriptionNewAcronymDef}{%
\edef\do@newglossaryentry{%
\noexpand\newglossaryentry{\the\glslabeltok}%
{%
type=\acronymtype,%
name={\noexpand
\acrnameformat{\the\glsshorttok}{\the\glslongtok}},%
access={\noexpand\glo@textaccess},%
sort={\the\glsshorttok},%
short={\the\glsshorttok},%
shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
long={\the\glslongtok},%
longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
text={\the\glsshorttok},%
textaccess={\the\glslongtok},%
plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
symbol={\noexpand\glo@text},%
symbolaccess={\noexpand\glo@textaccess},%
symbolplural={\noexpand\glo@plural},%
firstpluralaccess=\relax,
textaccess={\noexpand\glo@shortaccess},%
\the\glskeylisttok}%
}%
\let\org@gls@assign@firstpl\gls@assign@firstpl
\let\org@gls@assign@plural\gls@assign@plural
\let\org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
\org@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
\org@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
\org@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\org@gls@assign@firstpl
\let\gls@assign@plural\org@gls@assign@plural
\let\gls@assign@symbolplural\org@gls@assign@symbolplural
}

```

```

\renewcommand*{\FootnoteNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      sort={\the\glsshorttok},%
      text={\the\glsshorttok},%
      textaccess={\the\glslongtok},%
      access={\noexpand\@glo@textaccess},%
      plural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      description={\the\glslongtok},%
      descriptionplural={\the\glslongtok\noexpand\acrpluralsuffix},%
      \the\glskeylisttok
    }%
  }%
  \let\@org@gls@assign@plural\gls@assign@plural
  \let\@org@gls@assign@firstpl\gls@assign@firstpl
  \let\@org@gls@assign@descplural\gls@assign@descplural
  \def\gls@assign@firstpl##1##2{%
    \@@gls@expand@field{##1}{firstpl}{##2}%
  }%
  \def\gls@assign@plural##1##2{%
    \@@gls@expand@field{##1}{plural}{##2}%
  }%
  \def\gls@assign@descplural##1##2{%
    \@@gls@expand@field{##1}{descplural}{##2}%
  }%
  \@do@newglossaryentry
  \let\gls@assign@plural\@org@gls@assign@plural
  \let\gls@assign@firstpl\@org@gls@assign@firstpl
  \let\gls@assign@descplural\@org@gls@assign@descplural
}
\renewcommand*{\SmallNewAcronymDef}{%
  \edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\noexpand\acronymfont{\the\glsshorttok}},%
      access={\noexpand\@glo@symbolaccess},%
      sort={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      shortaccess={\glsdefaultshortaccess{\the\glslongtok}{\the\glsshorttok}},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\acrpluralsuffix},%

```

```

text={\noexpand\@glo@short},%
textaccess={\noexpand\@glo@shortaccess},%
plural={\noexpand\@glo@shortpl},%
first={\the\glslongtok},%
firstaccess=\relax,
firstplural={\the\glslongtok\noexpand\acrpluralsuffix},%
description={\noexpand\@glo@first},%
descriptionplural={\noexpand\@glo@firstplural},%
symbol={\the\glsshorttok},%
symbolaccess={\the\glslongtok},%
symbolplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
\the\glskeylisttok
}%
}%
\let\@org@gls@assign@firstpl\gls@assign@firstpl
\let\@org@gls@assign@plural\gls@assign@plural
\let\@org@gls@assign@descplural\gls@assign@descplural
\let\@org@gls@assign@symbolplural\gls@assign@symbolplural
\def\gls@assign@firstpl##1##2{%
  \@@gls@expand@field{##1}{firstpl}{##2}%
}%
\def\gls@assign@plural##1##2{%
  \@@gls@expand@field{##1}{plural}{##2}%
}%
\def\gls@assign@descplural##1##2{%
  \@@gls@expand@field{##1}{descplural}{##2}%
}%
\def\gls@assign@symbolplural##1##2{%
  \@@gls@expand@field{##1}{symbolplural}{##2}%
}%
\do@newglossaryentry
\let\gls@assign@firstpl\@org@gls@assign@firstpl
\let\gls@assign@plural\@org@gls@assign@plural
\let\gls@assign@descplural\@org@gls@assign@descplural
\let\gls@assign@symbolplural\@org@gls@assign@symbolplural
}
\newcommand*{\glsshortaccesskey}{\glsshortkey access}%
\newcommand*{\glsshortpluralaccesskey}{\glsshortpluralkey access}%
\newcommand*{\glslongaccesskey}{\glslongkey access}%
\newcommand*{\glslongpluralaccesskey}{\glslongpluralkey access}%
\newcommand*{\showglonameaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@access\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@textaccess\endcsname
}
\newcommand*{\showglopluralaccess}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@pluralaccess\endcsname
}
\newcommand*{\showglofirstaccess}[1]{%

```

```

\expandafter\show\csname glo@\glsdetoklabel{#1}@firstaccess\endcsname
}
\newcommand*{\showglofirstpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@firstpluralaccess\endcsname
}
\newcommand*{\showglosymbolaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolaccess\endcsname
}
\newcommand*{\showglosymbolpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@symbolpluralaccess\endcsname
}
\newcommand*{\showglodescaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descaccess\endcsname
}
\newcommand*{\showglodescpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@descpluralaccess\endcsname
}
\newcommand*{\showgloshortaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@shortaccess\endcsname
}
\newcommand*{\showgloshortpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@shortpluralaccess\endcsname
}
\newcommand*{\showglolongaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longaccess\endcsname
}
\newcommand*{\showglolongpluralaccess}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@longpluralaccess\endcsname
}

```

8.20 Rollback v4.49 (glossaries-babel-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-babel}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak

```

```

        Language modules need to be installed separately.\MessageBreak
        Please check on CTAN for a bundle called\MessageBreak
        ‘glossaries-CurrentTrackedLanguage’ or similar}%
    }%
}%
}%
{}%

```

8.21 Rollback v4.49 (glossaries-compatible-207-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-207}[2021/11/01 v4.49 (NLCT)]
\ifglxsindy
  \renewcommand*\GlsAddXdyAttribute[1]{%
    \edef\xdyattributes{\@xdyattributes ^^J \string"#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J}%
    (markup-locref
    :open \string"\string~n\string\setentrycounter
      {\noexpand\glscounter}%
      \expandafter\string\csname#1\endcsname
      \expandafter@gobble\string\{\string" ^^J
    :close \string"\expandafter@gobble\string\}\string" ^^J
    :attr \string"#1\string"))}
\fi
\renewcommand*\GlsAddXdyCounters[1]{%
  \GlossariesWarning{\string\GlsAddXdyCounters\space not available
    in compatibility mode.}%
}
\GlsAddXdyAttribute{glshyphenformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshyphennumber}
\GlsAddXdyAttribute{hyperterm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hyperterm}
\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@mkidxrangechar\@glsnumberformat\@nil
  \def\@glo@range{}%
  \expandafter\if\@glo@prefix(\relax
    \def\@glo@range{:open-range}%
  \else
    \expandafter\if\@glo@prefix\relax
    \def\@glo@range{:close-range}%
  \fi
  \fi
  \protected@edef\@glslocref{\theglentrycounter}%
  \@gls@checkmkidxchars\@glslocref
  \glossary[\csname glo@#1@type\endcsname]{%
    (indexentry :tkey (\csname glo@#1@index\endcsname)
      :locref \string"\@glslocref\string" %
      :attr \string"\@glo@suffix\string" \@glo@range
    )
  }%
\else
  \@set@glo@numformat\@glo@numfmt\@gls@counter\@glsnumberformat
  \glossary[\csname glo@#1@type\endcsname]{%
    \string\glossaryentry{\csname glo@#1@index\endcsname
      \@gls@encapchar\@glo@numfmt}{\theglentrycounter}}%
  \fi
}
\def\@set@glo@numformat#1#2#3{%
  \expandafter\@glo@check@mkidxrangechar#3\@nil
  \protected@edef#1{%
    \@glo@prefix setentrycounter[] {#2}%
    \expandafter\string\csname\@glo@suffix\endcsname
  }%
  \@gls@checkmkidxchars#1%
}

```

```

\ifglxindy
\def\writeist{%
  \openout\glswrite=\istfilename
  \write\glswrite{;; xindy style file created by the glossaries
    package in compatible-2.07 mode}%
  \write\glswrite{;; for document '\jobname' on
    \the\year-\the\month-\the\day}%
  \write\glswrite{^^J; required styles^^J}
  \for\@xdystyle:=\@xdyrequiredstyles\do{%
    \ifx\@xdystyle\@empty
    \else
      \protected@write\glswrite{{(require
        \string"\@xdystyle.xdy\string")}}%
    \fi
  }%
  \write\glswrite{^^J%
    ; list of allowed attributes (number formats)^^J}%
  \write\glswrite{(define-attributes ((\@xdyattributes)))}%
  \write\glswrite{^^J; user defined alphabets^^J}%
  \write\glswrite{\@xdyuseralphabets}%
  \write\glswrite{^^J; location class definitions^^J}%
  \protected@edef\@gls@roman{\@roman{0}\string"
    \string"roman-numbers-lowercase\string" :sep \string"}}%
  \@onelevel@sanitize\@gls@roman
  \edef\@tmp{\string" \string"roman-numbers-lowercase\string"
    :sep \string"}%
  \@onelevel@sanitize\@tmp
  \ifx\@tmp\@gls@roman
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (\string"roman-numbers-lowercase\string")
      :min-range-length \@glsminrange)}}%
  \else
    \write\glswrite{(define-location-class
      \string"roman-page-numbers\string"^^J\space\space\space
      (:sep "\@gls@roman")
      :min-range-length \@glsminrange)}}%
  \fi
  \write\glswrite{(define-location-class
    \string"Roman-page-numbers\string"^^J\space\space\space
    (\string"roman-numbers-uppercase\string")
    :min-range-length \@glsminrange)}}%
  \write\glswrite{(define-location-class
    \string"arabic-page-numbers\string"^^J\space\space\space
    (\string"arabic-numbers\string")
    :min-range-length \@glsminrange)}}%
  \write\glswrite{(define-location-class
    \string"alpha-page-numbers\string"^^J\space\space\space
    (\string"alpha\string")
    :min-range-length \@glsminrange)}}%

```

```

\write\glswrite{(define-location-class
  \string"Alpha-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string")
    :min-range-length \@glsmminrange)}%
\write\glswrite{(define-location-class
  \string"Appendix-page-numbers\string"^^J\space\space\space
  (\string"ALPHA\string"
    :sep \string"\@glAlphacompositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmminrange)}%
\write\glswrite{(define-location-class
  \string"arabic-section-numbers\string"^^J\space\space\space
  (\string"arabic-numbers\string"
    :sep \string"\glscpositor\string"
    \string"arabic-numbers\string")
    :min-range-length \@glsmminrange)}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{@xduyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
  :unverified )}%
\write\glswrite{(markup-crossref-list
  :class \string"see\string"^^J\space\space\space
  :open \string"\string\glseeformat\string"
  :close \string"{}\string")}%
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
  (\@xduylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
  :open \string"\string
  \glossarysection[\string\glossarytoctitle]{\string
  \glossarytitle}\string\glossarypreamble\string~n\string\begin
  {theglossary}\string\glossaryheader\string~n\string" ^^J\space
  \space\space:close \string"\expandafter\@gobble
  \string\%\string~n\string
  \end{theglossary}\string\glossarypostamble
  \string~n\string" ^^J\space\space\space
  :tree)}}%
\write\glswrite{(markup-letter-group-list
  :sep \string"\string\glsgroupskip\string~n\string")}%
\write\glswrite{(markup-indexentry
  :open \string"\string\relax \string\glresetentrylist
  \string~n\string")}%
\write\glswrite{(markup-locclass-list :open
  \string"\glsoopenbrace\string\glossaryentrynumbers
  \glsoopenbrace\string\relax\space \string"^^J\space\space\space
  :sep \string", \string"
  :close \string"\glsclosebrace\glsclosebrace\string")}%
\write\glswrite{(markup-locref-list

```

```

:sep \string"\string\delimN\space\string")}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string")}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\string~\string")}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\noist}
\else
\edef\@gls@actualchar{\string?}
\edef\@gls@encapchar{\string|}
\edef\@gls@levelchar{\string!}
\edef\@gls@quotechar{\string"}
\def\writeist{\relax
\openout\glswrite=\istfilename
\write\glswrite{\expandafter\@gobble\string\% makeindex style file
created by the glossaries package}
\write\glswrite{\expandafter\@gobble\string\% for document
'\jobname' on \the\year-\the\month-\the\day}
\write\glswrite{actual '\@gls@actualchar'}
\write\glswrite{encap '\@gls@encapchar'}
\write\glswrite{level '\@gls@levelchar'}
\write\glswrite{quote '\@gls@quotechar'}
\write\glswrite{keyword \string"\string\glossaryentry\string"}
\write\glswrite{preamble \string"\string\glossarysection[\string
\glossarytoctitle]{\string\glossarytitle}\string
\glossarypreamble\string\n\string\begin{theglossary}\string
\glossaryheader\string\n\string"}

```

```

\write\glswrite{postamble \string"\string%\string\n\string
  \end{theglossary}\string\glossarypostamble\string\n
  \string"}
\write\glswrite{group_skip \string"\string\glsgroupskip\string\n
  \string"}
\write\glswrite{item_0 \string"\string%\string\n\string"}
\write\glswrite{item_1 \string"\string%\string\n\string"}
\write\glswrite{item_2 \string"\string%\string\n\string"}
\write\glswrite{item_01 \string"\string%\string\n\string"}
\write\glswrite{item_x1
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{item_12 \string"\string%\string\n\string"}
\write\glswrite{item_x2
  \string"\string\relax \string\glsresetentrylist\string\n
  \string"}
\write\glswrite{delim_0 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_1 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_2 \string"\string\{\string
  \glossaryentrynumbers\string\{\string\relax \string"}
\write\glswrite{delim_t \string"\string\}\string}\string"}
\write\glswrite{delim_n \string"\string\delimN \string"}
\write\glswrite{delim_r \string"\string\delimR \string"}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
  \string"\string\glsgroupheading\string\{\string"}
\write\glswrite{heading_suffix
  \string"\string\}\string\relax
  \string\glsresetentrylist \string"}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glsglnumbers\string"}
\write\glswrite{page_compositor \string"glsglcompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
  \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
  \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\noist
}
\fi
\renewcommand*{\noist}{\let\writeist\relax}

```

8.22 Rollback v4.49 (glossaries-compatible-307-2021-11-01.sty)

Version 4.49 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-compatible-307}[2021/11/01 v4.49 (NLCT)]
\newcommand{\compatglossarystyle}[2]{%
  \ifcsundef{@glscompstyle@#1}%
  {%
    \csdef{@glscompstyle@#1}{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary compatibility style ‘#1’ is already defined}{}%
  }%
}
\compatglossarystyle{inline}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \glsinlinedopostchild
    \gls@inlinesep
    \def\glo@desc{##3}%
    \def\@no@post@desc{\nopostdesc}%
    \glsentryitem{##1}\glsinlinenameformat{##1}{##2}%
    \ifx\glo@desc\@no@post@desc
      \glsinlineemptydescformat{##4}{##5}%
    \else
      \ifstrempy{##3}%
        {\glsinlineemptydescformat{##4}{##5}}%
        {\glsinlinedescformat{##3}{##4}{##5}}%
      \fi
      \ifglshaschildren{##1}%
      {%
        \glsresetsubentrycounter
        \glsinlineparentchildseparator
        \def\gls@inlinesubsep{}%
        \def\gls@inlinepostchild{\glsinlinepostchild}%
      }%
      {}%
    \def\gls@inlinesep{\glsinlineseparator}%
  }%
  \renewcommand{\glossarysubentryfield}[6]{%
    \gls@inlinesubsep%
    \glsinlinesubnameformat{##2}{##3}%
    \glssubentryitem{##2}\glsinlinesubdescformat{##4}{##5}{##6}%
    \def\gls@inlinesubsep{\glsinlinesubseparator}%
  }%
}
\compatglossarystyle{list}{%
  \renewcommand*\glossaryentryfield}[5]{%
    \item[\glsentryitem{##1}\glstarget{##1}{##2}]
```

```

        ##3\glspostdescription\space ##5}%
\renewcommand*{\glossarysubentryfield}[6]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6.}%
}
\compatglossarystyle{listgroup}{%
    \csuse{@glscmpstyle@list}%
}%
\compatglossarystyle{listhypergroup}{%
    \csuse{@glscmpstyle@list}%
}%
\compatglossarystyle{altlist}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[\glentryitem{##1}\glstarget{##1}{##2}]{%
            \mbox{}\par\nobreak\@afterheading
            ##3\glspostdescription\space ##5}%
    }
    \renewcommand{\glossarysubentryfield}[6]{%
        \par
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6}%
    }%
\compatglossarystyle{altlistgroup}{%
    \csuse{@glscmpstyle@altlist}%
}%
\compatglossarystyle{altlisthypergroup}{%
    \csuse{@glscmpstyle@altlist}%
}%
\compatglossarystyle{listdotted}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[]\makebox[\glslistdottedwidth][l]{%
            \glentryitem{##1}\glstarget{##1}{##2}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##3}%
    }
    \renewcommand*{\glossarysubentryfield}[6]{%
        \item[]\makebox[\glslistdottedwidth][l]{%
            \glssubentryitem{##2}%
            \glstarget{##2}{##3}%
            \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}##4}%
    }%
\compatglossarystyle{sublistdotted}{%
    \csuse{@glscmpstyle@listdotted}%
    \renewcommand*{\glossaryentryfield}[5]{%
        \item[\glentryitem{##1}\glstarget{##1}{##2}]{%
    }%
\compatglossarystyle{long}{%
    \renewcommand*{\glossaryentryfield}[5]{%
        \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
    }
    \renewcommand*{\glossarysubentryfield}[6]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
    }

```

```

}%
\compatglossarystyle{longborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheader}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{longheaderborder}{%
  \csuse{@glscompstyle@long}%
}%
\compatglossarystyle{long3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\\}%
}%
\compatglossarystyle{long3colborder}{%
  \csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheader}{%
  \csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long3colheaderborder}{%
  \csuse{@glscompstyle@long3col}%
}%
\compatglossarystyle{long4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
}%
\compatglossarystyle{long4colheader}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{long4colheaderborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4col}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheader}{%
  \csuse{@glscompstyle@long4col}%
}%

```



```

\compatglossarystyle{altlong4colborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{altlong4colheaderborder}{%
  \csuse{@glscompstyle@long4col}%
}%
\compatglossarystyle{longragged}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
    \tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4\glspostdescription\space ##6%
    \tabularnewline}%
}%
\compatglossarystyle{longraggedborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheader}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longraggedheaderborder}{%
  \csuse{@glscompstyle@longragged}%
}%
\compatglossarystyle{longragged3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{longragged3colborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheader}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{longragged3colheaderborder}{%
  \csuse{@glscompstyle@longragged3col}%
}%
\compatglossarystyle{altlongragged4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%

```

```

\compatglossarystyle{altlongragged4colheader}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{altlongragged4colheaderborder}{%
  \csuse{@glscompstyle@altlong4col}%
}%
\compatglossarystyle{index}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \item\glstarget{##1}{\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
      \else
        \space(##4)%
      \fi
    \space ##3\glspostdescription \space ##5}%
  \renewcommand*{\glossarysubentryfield}[6]{%
    \ifcase##1\relax
      % level 0
      \item
    \or
      % level 1
      \subitem
    \glssubentryitem{##2}%
  \else
      % all other levels
      \subsubitem
    \fi
    \textbf{\glstarget{##2}{##3}}%
    \ifx\relax##5\relax
      \else
        \space(##5)%
      \fi
    \space##4\glspostdescription\space ##6}%
  }%
\compatglossarystyle{indexgroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{indexhypergroup}{%
  \csuse{@glscompstyle@index}%
}%
\compatglossarystyle{tree}{%
  \renewcommand{\glossaryentryfield}[5]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glstarget{##1}{\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
      \else
        \space(##4)%
      \fi
    }%

```

```

\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\textbf{\glstarget{##2}{##3}}%
\ifx\relax##5\relax
\else
\space(##5)%
\fi
\space##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treegroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treehypergroup}{%
\csuse{@glscmpstyle@tree}%
}%
\compatglossarystyle{treenoname}{%
\renewcommand{\glossaryentryfield}[5]{%
\hangindent0pt\relax
\parindent0pt\relax
\glssubentryitem{##1}\textbf{\glstarget{##1}{##2}}%
\ifx\relax##4\relax
\else
\space(##4)%
\fi
\space ##3\glspostdescription \space ##5\par}%
\renewcommand{\glossarysubentryfield}[6]{%
\hangindent##1\glstreeindent\relax
\parindent##1\glstreeindent\relax
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\glstarget{##2}{\strut}%
##4\glspostdescription\space ##6\par}%
}%
\compatglossarystyle{treenonamegroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{treenonamehypergroup}{%
\csuse{@glscmpstyle@treenoname}%
}%
\compatglossarystyle{alttree}{%
\renewcommand{\glossaryentryfield}[5]{%
\ifnum\@gls@prevlevel=0\relax
\else

```

```

        \settowidth{\glstreeindent}{\textbf{\@glswidestname\space}}%
        \hangindent\glstreeindent
        \parindent\glstreeindent
    \fi
    \makebox[0pt][r]{\makebox[\glstreeindent][l]{%
        \glssentryitem{##1}\textbf{\glstarget{##1}{##2}}}%
    \ifx\relax##4\relax
    \else
        (##4)\space
    \fi
    ##3\glspostdescription\space ##5\par
    \def\@gls@prevlevel{0}%
}%
\renewcommand{\glossarysubentryfield}[6]{%
    \ifnum##1=1\relax
        \glssubentryitem{##2}%
    \fi
    \ifnum\@gls@prevlevel=##1\relax
    \else
        \ifundefined{\@glswidestname\romannumeral##1}{%
            \settowidth{\gls@tmplen}{\textbf{\@glswidestname\space}}{%
            \settowidth{\gls@tmplen}{\textbf{%
                \csname \@glswidestname\romannumeral##1\endcsname\space}}}%
        \ifnum\@gls@prevlevel<##1\relax
            \setlength\glstreeindent\gls@tmplen
            \addtolength\glstreeindent\parindent
            \parindent\glstreeindent
        \else
            \ifundefined{\@glswidestname\romannumeral\@gls@prevlevel}{%
                \settowidth{\glstreeindent}{\textbf{%
                    \@glswidestname\space}}}%
                \settowidth{\glstreeindent}{\textbf{%
                    \csname \@glswidestname\romannumeral\@gls@prevlevel
                    \endcsname\space}}}%
            \addtolength\parindent{-\glstreeindent}%
            \setlength\glstreeindent\parindent
        \fi
    \fi
    \hangindent\glstreeindent
    \makebox[0pt][r]{\makebox[\gls@tmplen][l]{%
        \textbf{\glstarget{##2}{##3}}}%
    \ifx##5\relax\relax
    \else
        (##5)\space
    \fi
    ##4\glspostdescription\space ##6\par
    \def\@gls@prevlevel{##1}%
}%
}%
\compatglossarystyle{almtreeegroup}{%

```

```

\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{almtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolindex}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexgroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcolindexhypergroup}{%
\csuse{@glscompstyle@index}%
}%
\compatglossarystyle{mcoltree}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolindextreehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreename}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamegroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcoltreenamehypergroup}{%
\csuse{@glscompstyle@tree}%
}%
\compatglossarystyle{mcolalmtree}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreegroup}{%
\csuse{@glscompstyle@almtree}%
}%
\compatglossarystyle{mcolalmtreehypergroup}{%
\csuse{@glscompstyle@almtree}%
}%

\compatglossarystyle{superragged}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glstentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5%
\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4\glspostdescription\space ##6%
}

```

```

\tabularnewline}%
}%
\compatglossarystyle{superraggedborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheader}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superraggedheaderborder}{%
\csuse{@glscompstyle@superragged}%
}%
\compatglossarystyle{superragged3col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##6\tabularnewline}%
}%
\compatglossarystyle{superragged3colborder}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheader}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{superragged3colheaderborder}{%
\csuse{@glscompstyle@superragged3col}%
}%
\compatglossarystyle{altsuperragged4col}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\tabularnewline}%
\renewcommand*{\glossarysubentryfield}[6]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}##4 & ##5 & ##6\tabularnewline}%
}%
\compatglossarystyle{altsuperragged4colheader}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colborder}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{altsuperragged4colheaderborder}{%
\csuse{@glscompstyle@altsuperragged4col}%
}%
\compatglossarystyle{super}{%
\renewcommand*{\glossaryentryfield}[5]{%
\glentryitem{##1}\glstarget{##1}{##2} & ##3\glspostdescription\space ##5\\}%
\renewcommand*{\glossarysubentryfield}[6]{%
&

```

```

        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}##4\glspostdescription\space ##6\\}%
}%
\compatglossarystyle{superborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheader}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{superheaderborder}{%
  \csuse{@glscmpstyle@super}%
}%
\compatglossarystyle{super3col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##5\\}%
    \renewcommand*{\glossarysubentryfield}[6]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}##4 & ##6\\}%
    }%
\compatglossarystyle{super3colborder}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheader}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super3colheaderborder}{%
  \csuse{@glscmpstyle@super3col}%
}%
\compatglossarystyle{super4col}{%
  \renewcommand*{\glossaryentryfield}[5]{%
    \glssubentryitem{##1}\glstarget{##1}{##2} & ##3 & ##4 & ##5\\}%
    \renewcommand*{\glossarysubentryfield}[6]{%
      &
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}##4 & ##5 & ##6\\}%
    }%
\compatglossarystyle{super4colheader}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{super4colborder}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{super4colheaderborder}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{altsuper4col}{%
  \csuse{@glscmpstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheader}{%

```

```

\csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colborder}{%
\csuse{@glscompstyle@super4col}%
}%
\compatglossarystyle{altsuper4colheaderborder}{%
\csuse{@glscompstyle@super4col}%
}%

```

8.23 Rollback v4.49 (glossaries-polyglossia-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-polyglossia}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{tracklang}
\let\glsifusetranslator\@secondoftwo
\AnyTrackedLanguages
{%
\ForEachTrackedDialect{\this@dialect}{%
\IfTrackedLanguageFileExists{\this@dialect}%
{glossaries-}% prefix
{.ldf}%
}%
\RequireGlossariesLang{\CurrentTrackedTag}%
}%
{%
\PackageWarningNoLine{glossaries}%
{No language module detected for ‘\this@dialect’.\MessageBreak
Language modules need to be installed separately.\MessageBreak
Please check on CTAN for a bundle called\MessageBreak
‘glossaries-\CurrentTrackedLanguage’ or similar}%
}%
}%
}%
{}%

```

8.24 Rollback v4.49 (glossaries-prefix-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossaries-prefix}[2021/11/01 v4.49 (NLCT)]
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{glossaries}}
\ProcessOptions
\RequirePackage{glossaries}[=v4.49]
\define@key{glossentry}{prefixfirst}{\def\@glo@entryprefixfirst{#1}}%
\define@key{glossentry}{prefixfirstplural}{\def\@glo@entryprefixfirstplural{#1}}%
\define@key{glossentry}{prefix}{\def\@glo@entryprefix{#1}}%
\define@key{glossentry}{prefixplural}{\def\@glo@entryprefixplural{#1}}%
\appto\@gls@keymap{,%

```



```

{prefixfirst}{prefixfirst},%
{prefixfirstplural}{prefixfirstplural},%
{prefix}{prefix},%
{prefixplural}{prefixplural}%
}
\appto\@newglossaryentryprehook{%
  \def\@glo@entryprefix{}%
  \def\@glo@entryprefixplural{}%
  \let\@glo@entryprefixfirst\@gls@default@value
  \let\@glo@entryprefixfirstplural\@gls@default@value
}
\appto\@newglossaryentryposthook{%
  \gls@assign@field{}\@glo@label}{prefix}{\@glo@entryprefix}%
  \gls@assign@field{}\@glo@label}{prefixplural}{\@glo@entryprefixplural}%
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @prefix\endcsname}{\@glo@label}{prefixfirst}%
    {\@glo@entryprefixfirst}%
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\@glo@label @prefixplural\endcsname}{\@glo@label}%
    {prefixfirstplural}{\@glo@entryprefixfirstplural}%
}
\newcommand*{\glsentryprefixfirst}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixfirst}}
\newcommand*{\glsentryprefixfirstplural}[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prefixfirstplural}}
\newcommand*{\glsentryprefix}[1]{\csuse{glo@\glsdetoklabel{#1}@prefix}}
\newcommand*{\glsentryprefixplural}[1]{\csuse{glo@\glsdetoklabel{#1}@prefixplural}}
\newrobustcmd*{\Glsentryprefixfirst}[1]{%
  \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirst\endcsname}%
  \xmakefirstuc\@glo@text
}
\newrobustcmd*{\Glsentryprefixfirstplural}[1]{%
  \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixfirstplural\endcsname}%
  \xmakefirstuc\@glo@text
}
\newrobustcmd*{\Glsentryprefix}[1]{%
  \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefix\endcsname}%
  \xmakefirstuc\@glo@text
}
\newrobustcmd*{\Glsentryprefixplural}[1]{%
  \protected@edef\@glo@text{\csname glo@\glsdetoklabel{#1}@prefixplural\endcsname}%
  \xmakefirstuc\@glo@text
}
\newcommand*{\ifglshasprefix}[3]{%
  \ifcsemtty{glo@\glsdetoklabel{#1}@prefix}%
    {#3}%
    {#2}%
}
\newcommand*{\ifglshasprefixplural}[3]{%
  \ifcsemtty{glo@\glsdetoklabel{#1}@prefixplural}%
    {#3}%

```

```

    {#2}%
}
\newcommand*{\ifglshasprefixfirst}[3]{%
  \ifcempty{glo@glsdetoklabel{#1}@prefixfirst}%
  {#3}%
  {#2}%
}
\newcommand*{\ifglshasprefixfirstplural}[3]{%
  \ifcempty{glo@glsdetoklabel{#1}@prefixfirstplural}%
  {#3}%
  {#2}%
}
}
\providecommand{\@glsprefix@record@hook}[2]{%
  \ifdef\@glsxtr@record
  {\@glsxtr@record{#1}{#2}{glslink}}%
  {}%
}
\newcommand{\glsprefixsep}{}
\newrobustcmd{\pgls}{\@glshyp@opt\@pgls}
\newcommand*{\@pgls}[2][{}]{%
  \new@ifnextchar[%
  {\@pgls@{#1}{#2}}%
  {\@pgls@{#1}{#2}[]}%
}
\def\@pgls@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsoifexists{#2}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}{\glsentryprefix{#2}\glsprefixsep}{}%
    }%
    {%
      \ifglshasprefixfirst{#2}{\glsentryprefixfirst{#2}\glsprefixsep}{}%
    }%
    \@gl@{#1}{#2}[#3]%
  }%
}
\newrobustcmd{\pglsp1}{\@glshyp@opt\@pglsp1}
\newcommand*{\@pglsp1}[2][{}]{%
  \new@ifnextchar[%
  {\@pglsp1@{#1}{#2}}%
  {\@pglsp1@{#1}{#2}[]}%
}
\def\@pglsp1@#1#2[#3]{%
  \@glsprefix@record@hook{#1}{#2}%
  \glsoifexists{#2}%
  {%
    \ifglused{#2}%
    {%

```

```

        \ifglshasprefixplural{#2}{\glstryentryprefixplural{#2}\glsprefixsep}{}%
    }%
    {%
        \ifglshasprefixfirstplural{#2}%
            {\glstryentryprefixfirstplural{#2}\glsprefixsep}{}%
        }%
        \@glsp1@{#1}{#2}[#3]%
    }%
}
\newrobustcmd{\Pgls}{\@glshyp@opt\Pgls}
\newcommand*{\@Pgls}[2][{}]{%
    \new@ifnextchar[%
        {\@Pgls@{#1}{#2}}%
        {\@Pgls@{#1}{#2}[{}]}%
    }
}
\def\@Pgls@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glstrydoifexists{#2}%
    {%
        \ifglssused{#2}%
        {%
            \ifglshasprefix{#2}%
            {%
                \glstryentryprefix{#2}%
                \glsprefixsep
                \@glss@{#1}{#2}[#3]%
            }%
            {\@glss@{#1}{#2}[#3]}%
        }%
        {%
            \ifglshasprefixfirst{#2}%
            {%
                \glstryentryprefixfirst{#2}%
                \glsprefixsep
                \@glss@{#1}{#2}[#3]%
            }%
            {\@glss@{#1}{#2}[#3]}%
        }%
    }%
}
}
\newrobustcmd{\Pglspl}{\@glshyp@opt\Pglspl}
\newcommand*{\@Pglspl}[2][{}]{%
    \new@ifnextchar[%
        {\@Pglspl@{#1}{#2}}%
        {\@Pglspl@{#1}{#2}[{}]}%
    }
}
\def\@Pglspl@#1#2[#3]{%
    \@glsprefix@record@hook{#1}{#2}%
    \glstrydoifexists{#2}%
    {%

```

```

\ifglused{#2}%
{%
  \ifglshasprefixplural{#2}%
  {%
    \Glsentryprefixplural{#2}%
    \glsprefixsep
    \@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}%
  {%
    \ifglused{#2}%
    {%
      \ifglshasprefix{#2}%
      {\mfirstucMakeUppercase{\glentryprefix{#2}\glsprefixsep}}{%
    }%
    {%
      \ifglshasprefixfirst{#2}%
      {\mfirstucMakeUppercase{\glentryprefixfirst{#2}\glsprefixsep}}{%
    }%
    \@GLS@{#1}{#2}[#3]%
  }%
}
}
\newrobustcmd{\PGLSpl}{\@gls@hyp@opt\PGLSpl}
\newcommand*{\@PGLSpl}[2][{}]{%
  \new@ifnextchar[%
    {\@PGLSpl@{#1}{#2}}%
    {\@PGLSpl@{#1}{#2}[]}%
}
\def\@PGLSpl@#1#2[#3]{%

```

```

\@glsprefix@record@hook{#1}{#2}%
\glsdoifexists{#2}%
{%
  \ifglsused{#2}%
  {%
    \ifglschasprefixplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixplural{#2}\glsprefixsep}}{}%
  }%
  {%
    \ifglschasprefixfirstplural{#2}%
    {\mfirstucMakeUppercase{\glsentryprefixfirstplural{#2}\glsprefixsep}}{}%
  }%
  \@GLSpl@{#1}{#2}{#3}%
}%
}

```

8.25 Rollback v4.49 (glossary-hypernav-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-hypernav}[2021/11/01 v4.49 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\@glo@type]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \@glslink{\glsnavhyperlinkname{#1}{#2}}{#3}}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}[3][\@glo@type]{%
  \@glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@glsnavhypertarget}[3]{%
  \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \@gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@rerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \@ifundefined{\@gls@hypergroup@list@#1}{%
    \expandafter\xdef\csname @gls@hypergroup@list@#1\endcsname{#2}%
  }{%

```

```

\expandafter\let\expandafter\@gls@tmp
\csname @gls@hypergroupplist@#1\endcsname
\expandafter\xdef\csname @gls@hypergroupplist@#1\endcsname{%
\@gls@tmp,#2}%
}%
}
\newcommand*{\glsnavigation}{%
\def\@gls@between{}%
\ifcsundef\@gls@hypergroupplist@\@gls@type}%
{%
\def\@gls@list{}%
}%
{%
\expandafter\let\expandafter\@gls@list
\csname @gls@hypergroupplist@\@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\glshypernavsep
}%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glssymbolnav}{%
\glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
\glshypernavsep
\glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
\glshypernavsep
}

```

8.26 Rollback v4.49 (glossary-inline-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-inline}[2021/11/01 v4.49 (NLCT)]
\newglossarystyle{inline}{%
\renewenvironment{theglossary}%
{%
\def\gls@inlinesep{}%
\def\gls@inlinesubsep{}%
\def\gls@inlinepostchild{}%
}%
{\glspostinline}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsinlinedopostchild
\gls@inlinesep
\glsentryitem{##1}%

```

```

\glsinlinenameformat{##1}{%
  \glossentryname{##1}%
}%
\ifglshasdescsuppressed{##1}%
{%
  \glsinlineemptydescformat
  {%
    \glossentrysymbol{##1}%
  }%
  {%
    ##2%
  }%
}%
{%
  \ifglshasdesc{##1}%
  {\glsinlinedescformat{\glossentrydesc{##1}}{\glossentrysymbol{##1}}{##2}}%
  {\glsinlineemptydescformat{\glossentrysymbol{##1}}{##2}}%
}%
\ifglshaschildren{##1}%
{%
  \glsresetsubentrycounter
  \glsinlineparentchildseparator
  \def\gls@inlinesubsep{}%
  \def\gls@inlinepostchild{\glsinlinepostchild}%
}%
{}%
\def\gls@inlinesep{\glsinlineseparator}%
}%
\renewcommand{\subglossentry}[3]{%
  \gls@inlinesubsep%
  \glsinlinesubnameformat{##2}{%
    \glossentryname{##2}}%
  \glsesubentryitem{##2}%
  \glsinlinesubdescformat{\glossentrydesc{##2}}{\glossentrysymbol{##2}}{##3}%
  \def\gls@inlinesubsep{\glsinlinesubseparator}%
}%
\renewcommand*{\glsgroupskip}{}%
}
\newcommand*{\glsinlinedopostchild}{%
  \gls@inlinepostchild
  \def\gls@inlinepostchild{}%
}
\newcommand*{\glsinlineseparator}{;\space}
\newcommand*{\glsinlinesubseparator}{,\space}
\newcommand*{\glsinlineparentchildseparator}{:\space}
\newcommand*{\glsinlinepostchild}{}
\newcommand*{\glspostinline}{\glspostdescription\space}
\newcommand*{\glsinlinenameformat}[2]{\glstarget{#1}{#2}}
\newcommand*{\glsinlinedescformat}[3]{\space#1}
\newcommand*{\glsinlineemptydescformat}[2]{}

```

```

\newcommand*{\glsinlinesubnameformat}[2]{\glstarget{#1}{}}
\newcommand*{\glsinlinesubdescformat}[3]{#1}

```

8.27 Rollback v4.49 (glossary-list-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-list}[2021/11/01 v4.49 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glslistgroupheaderfmt}[1]{#1}
\newcommand*{\glslistnavigationitem}[1]{\item[#1]}
\newcommand*{\glslistinit}{%
  \ifdef\GetTitleStringDisableCommands
  {%
    \GetTitleStringSetup{expand}%
    \GetTitleStringDisableCommands{%
      \let\glsentryitem\@gobble
      \let\glstarget\@secondoftwo
      \let\glossentryname\glslistexpandedname
      \let\glslistgroupheaderfmt\@firstofone
      \let\glsgetgrouptitle\@firstofone
      \let\glsnavhypertarget\@secondoftwo
      \let\glsnavigation\relax
    }%
  }%
}%
}
\newcommand{\glslistexpandedname}[1]{%
  \glsunexpandedfieldvalue{#1}{name}%
}
\newglossarystyle{list}{%
  \renewenvironment{theglossary}{%
    {\glslistinit\begin{description}}{\end{description}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}]
    \glossentrydesc{##1}\glspostdescription\space ##2}%
  \renewcommand*{\subglossentry}[3]{%
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\space
    \glossentrydesc{##2}\glspostdescription\space ##3.}%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{listgroup}{%
  \setglossarystyle{list}%
}

```



```

\renewcommand*{\glsgroupheading}[1]{%
  \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{listhypergroup}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{altlist}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \mbox{}\par\nobreak\@afterheading
      \glossentrydesc{##1}\glspostdescription\space ##2}%
    \renewcommand{\subglossentry}[3]{%
      \par
      \glssubentryitem{##2}%
      \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space ##3}%
    }
\newglossarystyle{altlistgroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt{\glsgetgrouptitle{##1}}]}
\newglossarystyle{altlisthypergroup}{%
  \setglossarystyle{altlist}%
  \renewcommand*{\glossaryheader}{%
    \glslistnavigationitem{\glsnavigation}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item[\glslistgroupheaderfmt
      {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}]}]}
\newglossarystyle{listdotted}{%
  \setglossarystyle{list}%
  \renewcommand*{\glossentry}[2]{%
    \item[\makebox[\glslistdottedwidth][l]{%
      \glsentryitem{##1}%
      \glstarget{##1}{\glossentryname{##1}}%
      \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##1}}%
    \renewcommand*{\subglossentry}[3]{%
      \item[\makebox[\glslistdottedwidth][l]{%
        \glssubentryitem{##2}%
        \glstarget{##2}{\glossentryname{##2}}%
        \unskip\leaders\hbox to 2.9mm{\hss.}\hfill\strut}\glossentrydesc{##2}}%
      }
    }
\newlength\glslistdottedwidth
\setlength{\glslistdottedwidth}{.5\hsize}
\newglossarystyle{sublistdotted}{%
  \setglossarystyle{listdotted}%
  \renewcommand*{\glossentry}[2]{%

```

```

\item[\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}}}%
}

```

8.28 Rollback v4.49 (glossary-long-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-long}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{long}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[lp{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longborder}{%
  \setglossarystyle{long}%
  \renewenvironment{theglossary}{%
    \begin{longtable}[llp{glsdescwidth}]{\end{longtable}}%
    \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
  }
\newglossarystyle{longheader}{%
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname\tabularnewline\endhead}%
  }

```

```

\newglossarystyle{longheaderborder}{%
  \setglossarystyle{longborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\hline
    \endhead
    \hline\endfoot}%
  }
\newglossarystyle{long3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}\lp{\glstdescwidth}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsgnognroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long3colborder}{%
  \setglossarystyle{long3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}\{ll|p{\glstdescwidth}|p{\glspagelistwidth}|\}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long3colheader}{%
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
  }
\newglossarystyle{long3colheaderborder}{%
  \setglossarystyle{long3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
  }

```

```

}
\newglossarystyle{long4col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} &
    ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{long4colheader}{%
  \setglossarystyle{long4col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{long4colborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{long4colheaderborder}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|l|l|l|l|l|}}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries\entryname&\bfseries\descriptionname&
    \bfseries \symbolname&
    \bfseries\pagelistname\tabularnewline\hline\endhead
    \hline\endfoot}%
}

```

```

\newglossarystyle{altlong4col}{%
  \setglossarystyle{long4col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheader}{%
  \setglossarystyle{long4colheader}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colborder}{%
  \setglossarystyle{long4colborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
    {\end{longtable}}%
}
\newglossarystyle{altlong4colheaderborder}{%
  \setglossarystyle{long4colheaderborder}%
  \renewenvironment{theglossary}%
    {\begin{longtable}{|lp{\glsgdescwidth}|lp{\glspagelistwidth}|}%
    {\end{longtable}}%
}
}

```

8.29 Rollback v4.49 (glossary-longbooktabs-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longbooktabs}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{booktabs}
\RequirePackage{glossary-long}[=v4.46]
\RequirePackage{glossary-longragged}[=v4.46]
\newglossarystyle{long-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long}%
  \renewcommand*{\glossaryheader}{%
    \toprule \bfseries \entryname & \bfseries
    \descriptionname\tabularnewline\midrule\endhead
    \bottomrule\endfoot}%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
  \fi
}
\newglossarystyle{long3col-booktabs}{%
  \glspatchLToutput
  \setglossarystyle{long3col}%
  \renewcommand*{\glossaryheader}{%

```

```

        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
\ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
\else
    \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
\fi
}
\newglossarystyle{long4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long4col}%
    \renewcommand*{\glossaryheader}{%
        \toprule \bfseries \entryname &
        \bfseries \descriptionname &
        \bfseries \symbolname &
        \bfseries \pagelistname
        \tabularnewline\midrule\endhead
        \bottomrule\endfoot}%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{\glspenaltygroupskip}%
    \fi
}
\newglossarystyle{altlong4col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long4col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{lp{\glsgdescwidth}lp{\glspagelistwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{longragged-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{longragged3col-booktabs}{%
    \glspatchLToutput
    \setglossarystyle{long3col-booktabs}%
    \renewenvironment{theglossary}%
        {\begin{longtable}{l>{\raggedright}p{\glsgdescwidth}%
            >{\raggedright}p{\glspagelistwidth}}}%
        {\end{longtable}}%
}
\newglossarystyle{altlongragged4col-booktabs}{%

```

```

\glspatchLToutput
\setglossarystyle{altlong4col-booktabs}%
\renewenvironment{theglossary}%
  {\begin{longtable}{l>{\raggedright}p{\glsdescwidth}l%
    >{\raggedright}p{\glspagelistwidth}}}%
  {\end{longtable}}%
}
\newcommand*{\glsLTpenaltycheck}{%
  \ifnum\outputpenalty=-50\vskip-\normalbaselineskip\relax\fi
}
\newcommand{\glspenaltygroupskip}{%
  \noalign{\penalty-50\vskip\normalbaselineskip}}
\let\@gls@org@LT@output\LT@output
\newcommand*{\glsrestoreLToutput}{\let\LT@output\@gls@org@LT@output}
\newcommand*{\glspatchLToutput}{%
  \renewcommand*{\LT@output}{%
    \ifnum\outputpenalty <-\@Mi
      \ifnum\outputpenalty > -\LT@end@pen
        \LT@err{floats and marginpars not allowed in a longtable}\@ehc
      \else
        \setbox\z@\vbox{\unvbox\@cclv}%
        \ifdim \ht\LT@lastfoot>\ht\LT@foot
          \dimen@ \pagegoal
          \advance\dimen@-\ht\LT@lastfoot
          \ifdim\dimen@<\ht\z@
            \setbox\@cclv\vbox{\unvbox\z@\copy\LT@foot\vss}%
            \@makecol
            \@outputpage
            \setbox\z@\vbox{\box\LT@head\glsLTpenaltycheck}%
          \fi
          \fi
          \global\@colroom\@colht
          \global\vsizel\@colht
          {\unvbox\z@\box\ifvoid\LT@lastfoot\LT@foot\else\LT@lastfoot\fi}%
        \fi
      \else
        \setbox\@cclv\vbox{\unvbox\@cclv\copy\LT@foot\vss}%
        \@makecol
        \@outputpage
        \global\vsizel\@colroom
        \copy\LT@head
        \glsLTpenaltycheck
        \nobreak
      \fi
    }%
  }
}

```

8.30 Rollback v4.49 (glossary-longragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-longragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{longtable}
\@ifundefined{glsdescwidth}{%
  \newlength{glsdescwidth}
  \setlength{glsdescwidth}{0.6\hsize}
}{%}
\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{longragged}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l>{\raggedright}p{glsdescwidth}]}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}%
    \glspostdescription\space ##3%
    \tabularnewline
  }%
  \ifglsnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & \tabularnewline}%
  \fi
}
\newglossarystyle{longraggedborder}{%
  \setglossarystyle{longragged}%
  \renewenvironment{theglossary}{%
    \begin{longtable}{|l|>{\raggedright}p{glsdescwidth}||}%
    {\end{longtable}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longraggedheader}{%
  \setglossarystyle{longragged}%
  \renewcommand*{\glossaryheader}{%
    \bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\endhead}%
}

```



```

}
\newglossarystyle{longraggedheaderborder}{%
  \setglossarystyle{longraggedborder}%
  \renewcommand*{\glossaryheader}{%
    \hline\bfseries \entryname & \bfseries \descriptionname
    \tabularnewline\hline
    \endhead
    \hline\endfoot}%
}
\newglossarystyle{longragged3col}{%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l>{\raggedright}p{\glsdescwidth}%
      >{\raggedright}p{\glspagelistwidth}}}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
  \fi
}
\newglossarystyle{longragged3colborder}{%
  \setglossarystyle{longragged3col}%
  \renewenvironment{theglossary}%
    {\begin{longtable}[l|l|>{\raggedright}p{\glsdescwidth}|%
      >{\raggedright}p{\glspagelistwidth}|}%
    {\end{longtable}}}%
  \renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{longragged3colheader}{%
  \setglossarystyle{longragged3col}%
  \renewcommand*{\glossaryheader}{%
    \bfseries\entryname&\bfseries\descriptionname&
    \bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{longragged3colheaderborder}{%
  \setglossarystyle{longragged3colborder}%
  \renewcommand*{\glossaryheader}{%
    \hline

```

```

\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}
\newglossarystyle{altlongragged4col}{%
\renewenvironment{theglossary}%
{\begin{longtable}[l>\raggedright]p{\glsdescwidth}l%
>\raggedright}p{\glspagelistwidth}}}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
\glossentrydesc{##1} & \glossentrysymbol{##1} &
##2\tabularnewline
}%
\renewcommand{\subglossentry}[3]{%
&
\glssubentryitem{##2}%
\glstarget{##2}{\strut}\glossentrydesc{##2} &
\glossentrysymbol{##2} & ##3\tabularnewline
}%
\ifglsnogroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{ & & \tabularnewline}%
\fi
}
\newglossarystyle{altlongragged4colheader}{%
\setglossarystyle{altlongragged4col}%
\renewenvironment{theglossary}%
{\begin{longtable}[l>\raggedright]p{\glsdescwidth}l%
>\raggedright}p{\glspagelistwidth}}}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{%
\bfseries\entryname&\bfseries\descriptionname&
\bfseries \symbolname&
\bfseries\pagelistname\tabularnewline\endhead}%
}
\newglossarystyle{altlongragged4colborder}{%
\setglossarystyle{altlongragged4col}%
\renewenvironment{theglossary}%
{\begin{longtable}[|l>\raggedright]p{\glsdescwidth}|l|%
>\raggedright}p{\glspagelistwidth}|}%
{\end{longtable}}}%
\renewcommand*{\glossaryheader}{\hline\endhead\hline\endfoot}%
}
\newglossarystyle{altlongragged4colheaderborder}{%
\setglossarystyle{altlongragged4col}%
\renewenvironment{theglossary}%

```

```

        {\begin{longtable}{|l|>{\raggedright}p{\glsdescwidth}|l|}%
          >{\raggedright}p{\glspagelistwidth}|}}%
        {\end{longtable}}}%
\renewcommand*{\glossaryheader}{%
\hline\bfseries\entryname&\bfseries\descriptionname&
\bfseries \symbolname&
\bfseries\pagelistname\tabularnewline\hline\endhead
\hline\endfoot}%
}

```

8.31 Rollback v4.49 (glossary-mcols-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-mcols}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.49]
\providecommand{\indexspace}{%
\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
\setglossarystyle{index}%
\renewenvironment{theglossary}{%
{%
\begin{multicols}{\glsmcols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem
\let\subitem\glstreesubitem
\let\subsubitem\glstreesubsubitem
}%
{\end{multicols}}}%
}
\newglossarystyle{mcolindexgroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
\setglossarystyle{mcolindex}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}}{\glsgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
\setglossarystyle{index}%

```

```

\renewenvironment{theglossary}%
{
  \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
  \setlength{\parindent}{0pt}%
  \setlength{\parskip}{0pt plus 0.3pt}%
  \let\item\glstreeitem}%
{\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
  \item\glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
  \indexspace}%
}
\newglossarystyle{mcoltree}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreegroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
  \setglossarystyle{mcoltree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreespannav}{%
  \setglossarystyle{tree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
    \indexspace}%
}

```

```

\newglossarystyle{mcoltreenoname}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
  \setglossarystyle{mcoltreenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
  \setglossarystyle{treenoname}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
    \setlength{\parindent}{0pt}%
    \setlength{\parskip}{0pt plus 0.3pt}%
  }%
  {\end{multicols}}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newglossarystyle{mcolalttree}{%
  \setglossarystyle{alttree}%
  \renewenvironment{theglossary}%
  {%
    \begin{multicols}{\glsmcols}
    \def\@gls@prevlevel{-1}%
    \mbox{}\par
  }%
  {\par\end{multicols}}%
}
\newglossarystyle{mcolalttreegroup}{%
  \setglossarystyle{mcolalttree}%

```

```

\renewcommand{\glsgroupheading}[1]{\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgrouptitle{##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}{\glsgrouptitle{##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{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}

```

8.32 Rollback v4.49 (glossary-super-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-super}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{supertabular}
\@ifundefined{glsdescwidth}{%
\newlength{glsdescwidth}
\setlength{glsdescwidth}{0.6\hspace}
}{}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{super}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}}}%
    {\end{supertabular}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3\tabularnewline
  }%
  \ifglsgnogroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|lp{\glstdescwidth}|}}%
    {\end{supertabular}}%
}
\newglossarystyle{superheader}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries \entryname &
      \bfseries\descriptionname\tabularnewline}%
    \tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}}}%
    {\end{supertabular}}%
}
\newglossarystyle{superheaderborder}{%
  \setglossarystyle{super}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries \entryname &
      \bfseries \descriptionname\tabularnewline\hline}%
    \tabletail{\hline}}%
    {\begin{supertabular}{|lp{\glstdescwidth}|}}%

```

```

        {\end{supertabular}}}%
    }
\newglossarystyle{super3col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glsentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super3colborder}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3colheader}{%
  \setglossarystyle{super3col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glsdescwidth}p{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super3colheaderborder}{%
  \setglossarystyle{super3colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname&\bfseries\descriptionname&
      \bfseries\pagelistname\tabularnewline\hline}%
    {\tabletail{\hline}%
    {\begin{supertabular}{|l|p{\glsdescwidth}|p{\glspagelistwidth}|}}%
    {\end{supertabular}}}%
}

```



```

\newglossarystyle{super4col}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{l|l|l|l}}{%
      \end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glosentry}[2]{%
    \glstryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1} &
    \glossentrysymbol{##1} & ##2\tabularnewline
  }%
  \renewcommand{\subglosentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2} &
    \glossentrysymbol{##2} & ##3\tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
  \fi
}
\newglossarystyle{super4colheader}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}%
    {\tabletail{}}%
    {\begin{supertabular}{l|l|l|l}}{%
      \end{supertabular}}}%
}
\newglossarystyle{super4colborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|l|l|l|l|}}%
    {\end{supertabular}}}%
}
\newglossarystyle{super4colheaderborder}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    {\tabletail{\hline}%
    {\begin{supertabular}{|l|l|l|l|}}%
    {\end{supertabular}}}%
}

```

```

}
\newglossarystyle{altsuper4col}{%
  \setglossarystyle{super4col}%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheader}{%
  \setglossarystyle{super4colheader}%
  \renewenvironment{theglossary}%
    {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline}\tabletail{}}%
    {\begin{supertabular}{lp{\glstdescwidth}lp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colborder}{%
  \setglossarystyle{super4colborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    \begin{supertabular}%
      {\lllp{\glstdescwidth}\lllp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}
\newglossarystyle{altsuper4colheaderborder}{%
  \setglossarystyle{super4colheaderborder}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {\lllp{\glstdescwidth}\lllp{\glspagelistwidth}}}%
    {\end{supertabular}}}%
}

```

8.33 Rollback v4.49 (glossary-superragged-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-superragged}[2021/11/01 v4.49 (NLCT)]
\RequirePackage{array}
\RequirePackage{supertabular}
\@ifundefined{glstdescwidth}{%
  \newlength{glstdescwidth}
  \setlength{glstdescwidth}{0.6\hspace}
}{%
}

```

```

\@ifundefined{glspagelistwidth}{%
  \newlength{glspagelistwidth}
  \setlength{glspagelistwidth}{0.1\hsize}
}{%}
\newglossarystyle{superragged}{%
  \renewenvironment{theglossary}%
    {\tablehead{}\tabletail{}}%
    {\begin{supertabular}{l>{\raggedright}p{\glsdescwidth}}}%
    {\end{supertabular}}}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupeheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \glstentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
    \glossentrydesc{##1}\glspostdescription\space ##2%
    \tabularnewline
  }%
  \renewcommand{\subglossentry}[3]{%
    &
    \glssubentryitem{##2}%
    \glstarget{##2}{\strut}\glossentrydesc{##2}\glspostdescription\space
    ##3%
    \tabularnewline
  }%
  \ifglsgroupskip
    \renewcommand*{\glsgroupskip}{}%
  \else
    \renewcommand*{\glsgroupskip}{& \tabularnewline}%
  \fi
}
\newglossarystyle{superraggedborder}{%
  \setglossarystyle{superragged}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline}\tabletail{\hline}%
    {\begin{supertabular}{|l|>{\raggedright}p{\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
}%
\ifglsgroupskip
\renewcommand*{\glsgroupskip}{}%
\else
\renewcommand*{\glsgroupskip}{& & \tabularnewline}%
\fi
}
\newglossarystyle{superragged3colborder}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\hline}\tabletail{\hline}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}||%
>{\raggedright}p{\glspagelistwidth}||}%
{\end{supertabular}}}%
}
\newglossarystyle{superragged3colheader}{%
\setglossarystyle{superragged3col}%
\renewenvironment{theglossary}%
{\tablehead{\bfseries\entryname&\bfseries\descriptionname&
\bfseries\pagelistname\tabularnewline}\tabletail{}%
\begin{supertabular}{|l|>{\raggedright}p{\glsdescwidth}%
>{\raggedright}p{\glspagelistwidth}}}%
{\end{supertabular}}}%
}
\newglossarystyle{superragged3colheaderborder}{%
\setglossarystyle{superragged3colborder}%
\renewenvironment{theglossary}%
{\tablehead{\hline

```

```

        \bfseries\entryname&\bfseries\descriptionname&
        \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}{|l|>\raggedright}p{\glsgdescwidth}|%
        >\raggedright}p{\glspagelistwidth}|}%
    \end{supertabular}}%
}
\newglossarystyle{altsuperragged4col}{%
    \renewenvironment{theglossary}%
        {\tablehead{} \tabletail{}}%
        \begin{supertabular}{|l>\raggedright}p{\glsgdescwidth}l%
            >\raggedright}p{\glspagelistwidth}|}%
        {\end{supertabular}}%
    \renewcommand*{\glossaryheader}{}%
    \renewcommand*{\glsgroupheading}[1]{}%
    \renewcommand{\glossentry}[2]{%
        \glssentryitem{##1}\glstarget{##1}{\glossentryname{##1}} &
        \glossentrydesc{##1} &
        \glossentrysymbol{##1} & ##2\tabularnewline
    }%
    \renewcommand{\subglossentry}[3]{%
        &
        \glssubentryitem{##2}%
        \glstarget{##2}{\strut}\glossentrydesc{##2} &
        \glossentrysymbol{##2} & ##3\tabularnewline
    }%
    \ifglsgroupskip
        \renewcommand*{\glsgroupskip}{}%
    \else
        \renewcommand*{\glsgroupskip}{& & \tabularnewline}%
    \fi
}
\newglossarystyle{altsuperragged4colheader}{%
    \setglossarystyle{altsuperragged4col}%
    \renewenvironment{theglossary}%
        {\tablehead{\bfseries\entryname&\bfseries\descriptionname&
            \bfseries\symbolname &
            \bfseries\pagelistname\tabularnewline}\tabletail{}}%
        \begin{supertabular}{|l>\raggedright}p{\glsgdescwidth}l%
            >\raggedright}p{\glspagelistwidth}|}%
        {\end{supertabular}}%
}
\newglossarystyle{altsuperragged4colborder}{%
    \setglossarystyle{altsuper4col}%
    \renewenvironment{theglossary}%
        {\tablehead{\hline}\tabletail{\hline}%
        \begin{supertabular}%
            { |l|>\raggedright}p{\glsgdescwidth}|l|%
            >\raggedright}p{\glspagelistwidth}|}%
        {\end{supertabular}}%
}

```

```

}
\newglossarystyle{altsuperragged4colheaderborder}{%
  \setglossarystyle{altsuperragged4col}%
  \renewenvironment{theglossary}%
    {\tablehead{\hline
      \bfseries\entryname &
      \bfseries\descriptionname &
      \bfseries\symbolname &
      \bfseries\pagelistname\tabularnewline\hline}%
    \tabletail{\hline}%
    \begin{supertabular}%
      {||>{\raggedright}p{\glsdescwidth}|||%
        >{\raggedright}p{\glspagelistwidth}||}%
    {\end{supertabular}}%
}

```

8.34 Rollback v4.49 (glossary-tree-2021-11-01.sty)

Version 4.49 preserved for rollback.

```

\ProvidesPackage{glossary-tree}[2021/11/01 v4.49 (NLCT)]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem
{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%

```

```

\item\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
}%
\renewcommand{\subglossentry}[3]{%
\ifcase##1\relax
% level 0
\item
\or
% level 1
\subitem
\glssubentryitem{##2}%
\else
% all other levels
\subsubitem
\fi
\glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
\ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
\glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
\setglossarystyle{index}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
\indexspace
}%
}
\newglossarystyle{indexhypergroup}{%
\setglossarystyle{index}%
\renewcommand*{\glossaryheader}{%
\item\glstreenavigationfmt{\glsnavigation}\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{tree}{%
\renewenvironment{theglossary}%
{\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}}%
{}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\hangindent0pt\relax
\parindent0pt\relax
\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
\ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
\glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par

```

```

}%
\renewcommand{\subglossentry}[3]{%
  \hangindent##1\glstreeindent\relax
  \parindent##1\glstreeindent\relax
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
  \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
  \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{treegroup}{%
  \setglossarystyle{tree}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par
    \indexspace}%
}
\newglossarystyle{treehypergroup}{%
  \setglossarystyle{tree}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glshypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newlength\glstreeindent
\setlength{\glstreeindent}{10pt}
\newglossarystyle{treenoname}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}}%
    {}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand{\glossentry}[2]{%
    \hangindent0pt\relax
    \parindent0pt\relax
    \glssentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi

```



```

\glstarget{##2}{\strut}%
\glossentrydesc{##2}\glspostdescription\space##3\par
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
\setglossarystyle{treenoname}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt
{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
\setglossarystyle{treenoname}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt
{\glslshhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newrobustcmd*{\glswidesttoplevelname}[1][\@glo@types]{%
\dimen@=0pt\relax
\gls@tmplen=0pt\relax
\forall glossaries[#1]{\@gls@type}%
{%
\forall glsentries[\@gls@type]{\@glo@label}%
{%
\ifglshasparent{\@glo@label}%
{}}%
{%
\settowidth{\dimen@}%
{\glstreenamefmt{\glstryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@glswidestname}{glo@glsetoklabel{\@glo@label}@name}%
\fi
}%
}%
}%
}
\newcommand*{\glswidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][l]{#2}%
}
\newglossarystyle{alttree}{%

```

```

\renewenvironment{theglossary}%
{
  \def\@gls@prevlevel{-1}%
  \mbox{}\par}%
{\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
  \ifnum\@gls@prevlevel=0\relax
  \else
    \settowidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
  \fi
  \hangindent\glstreeindent
  \parindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
    \glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
  \ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
  \glossentrydesc{##1}\glspostdescription \space ##2\par
  \def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
  \ifnum##1=1\relax
    \glssubentryitem{##2}%
  \fi
  \ifnum\@gls@prevlevel=##1\relax
  \else
    \@ifundefined{@glswidestname\romannumeral##1}{%
      \settowidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
        \settowidth{\gls@tmplen}{\glstreenamefmt{%
          \csname @glswidestname\romannumeral##1\endcsname\space}}}%
    \ifnum\@gls@prevlevel<##1\relax
      \setlength\glstreeindent\gls@tmplen
      \addtolength\glstreeindent\parindent
      \parindent\glstreeindent
    \else
      \@ifundefined{@glswidestname\romannumeral\@gls@prevlevel}{%
        \settowidth{\glstreeindent}{\glstreenamefmt{%
          \@glswidestname\space}}{%
          \settowidth{\glstreeindent}{\glstreenamefmt{%
            \csname @glswidestname\romannumeral\@gls@prevlevel
              \endcsname\space}}}%
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
      \fi
    \fi
  \hangindent\glstreeindent
  \makebox[Opt][r]{\glstreenamebox{\gls@tmplen}{%
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
  \ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
  \glossentrydesc{##2}\glspostdescription\space ##3\par
  \def\@gls@prevlevel{##1}%

```

```

}%
\renewcommand*{\glsgroupskip}{\ifglsgnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
\setglossarystyle{almtree}%
\renewcommand{\glsgroupheading}[1]{\par
\def\@glsprevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
\par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
\setglossarystyle{almtree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@glsprevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsgroupnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@glsprevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt
{\glsgroupnavigationtarget{##1}\glsgroupnavigationtitle{##1}}\par
\indexspace}}

```

8.35 Rollback v4.52 (glossaries-2022-11-03.sty)

```

\NeedsTeXFormat{LaTeX2e}
\DeclareRelease{v4.46}{2020-03-19}{glossaries-2020-03-19.sty}
\DeclareRelease{v4.49}{2021-11-01}{glossaries-2021-11-01.sty}
\DeclareCurrentRelease{v4.52}{2022-11-03}
\ProvidesPackage{glossaries}[2022/11/03 v4.52 (NLCT)]
\RequirePackage{ifthen}
\RequirePackage{xkeyval}[2006/11/18]
\RequirePackage{mfirstuc}
\newcommand{\glssentencecase}[1]{\makefirstuc{##1}}
\ExplSyntaxOn
\newcommand{\glslowercase}[1]{\text_lowercase:n{##1}}
\newcommand{\glssupercase}[1]{\text_uppercase:n{##1}}
\ifdef\MFUsentencecase
{%
\newcommand{\glsmfuexcl}[1]{\MFUexcl{##1}}
\newcommand{\glsmfublocker}[1]{\MFUblocker{##1}}
\newcommand{\glsmfuaddmap}[2]{\MFUaddmap{##1}{##2}}
}
{

```

```

\RequirePackage{textcase}
\providecommand{\MFUsentencecase}[1]{\text_titlecase_first:n{#1}}
\providecommand{\glsmfuexcl}[1]{
  \tl_if_in:NnF \l_text_case_exclude_arg_tl {#1}
  {
    \tl_put_right:Nn \l_text_case_exclude_arg_tl {#1}
  }
}
\renewcommand*{\mfirstucMakeUppercase}{\MakeTextUppercase}
\newcommand{\glsmfublocker}[1]{\glsmfuexcl{#1}}
\newcommand{\glsmfuaddmap}[2]{\glsmfuexcl{#1}\glsmfublocker{#2}}
}
\ExplSyntaxOff
\RequirePackage{xfor}
\RequirePackage{datatool-base}
\RequirePackage{amsgen}
\RequirePackage{etoolbox}
\newif\if@gls@docloaded
\@ifpackageloaded{doc}{%
  {%
    \@gls@docloadedtrue
  }%
  {%
    \@ifclassloaded{nlctdoc}{\@gls@docloadedtrue}{\@gls@docloadedfalse}%
  }
}
\if@gls@docloaded
  \let\glsorg@theglossary\theglossary
  \let\glsorg@endtheglossary\endtheglossary
  \let\glsorg@PrintChanges\PrintChanges
  \renewcommand{\PrintChanges}{%
    \begingroup
      \let\theglossary\glsorg@theglossary
      \let\endtheglossary\glsorg@endtheglossary
      \glsorg@PrintChanges
    \endgroup
  }
\fi
\newif\if@gls@debug
\define@choicekey{glossaries.sty}{debug}[\gls@debug@val\gls@debug@nr]{%
  {true,false,showtargets,showaccsupp}[true]{%
    \ifcase\gls@debug@nr\relax
      % debug=true
      \@gls@debugtrue
      \renewcommand*{\GlossariesWarning}[1]{%
        \PackageWarning{glossaries}{##1}%
      }%
      \renewcommand*{\GlossariesWarningNoLine}[1]{%
        \PackageWarningNoLine{glossaries}{##1}%
      }%
      \let\@glsshowtarget\@gobble

```

```

\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\or
% debug=false
\@gls@debugfalse
\let\@glsshowtarget\@gobble
\let\@glsshowaccsupp\@gobblethree
\PackageInfo{glossaries}{debug mode OFF}%
\or
% debug=showtargets
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowtarget}{\@glsshowtarget}%
\or
% debug=showaccsupp
\@gls@debugtrue
\renewcommand*\GlossariesWarning}[1]{%
\PackageWarning{glossaries}{##1}%
}%
\renewcommand*\GlossariesWarningNoLine}[1]{%
\PackageWarningNoLine{glossaries}{##1}%
}%
\PackageInfo{glossaries}{debug mode ON (nowarn option disabled)}%
\renewcommand{\@glsshowaccsupp}{\glsshowaccsupp}%
\fi
}
\newcommand*\glsshowtarget}[1]{%
\ifmmode
\glsshowtargetinner{#1}%
\else
\ifinner
\glsshowtargetinner{#1}%
\else
\glsshowtargetouter{#1}%
\fi
\fi
}
\newcommand*\glsshowtargetfonttext}[1]{%
\ifmmode \nfss@text{\glsshowtargetfont #1}\else {\glsshowtargetfont #1}\fi
}
\newcommand*\glsshowtargetinner}[1]{\glsshowtargetfonttext{[#1]}}
\newcommand*\glsshowtargetouter}[1]{%
\glsshowtargetsymbol\marginpar{\glsshowtargetsymbol\glsshowtargetfont #1}}
\newcommand*\glsshowtargetsymbol}{\tiny$\triangleright$}
\newcommand*\glsshowtargetfont}{\ttfamily\footnotesize}

```

```

\newcommand*{\@glsshowtarget}[1]{%
\newcommand*{\glsdoshowtarget}[2]{\@glsshowtarget{#1}#2}
\newrobustcmd*{\@@glsshowtarget}[1]{%
  \begingroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{%
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \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]{}
\glsesclocationsfalse
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{} }
\IfFileExists{supertabular.sty}{%
  \newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}}{%
  \newcommand*{\@gls@loadsuper}{} }
\@ifpackageloaded{xtab}{\renewcommand*{\@gls@loadsuper}{} }{}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{} }
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}}
\@gls@declareoption{nolist}{%
  \renewcommand*{\@gls@loadlist}{%
  \ifdefstring{\@glossary@default@style}{list}%
  {\let\@glossary@default@style\relax}%
  }%
}%
}%
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{} }
\@gls@declareoption{nostyles}{%
  \renewcommand*{\@gls@loadlong}{}%
  \renewcommand*{\@gls@loadsuper}{}%
  \renewcommand*{\@gls@loadlist}{}%
  \renewcommand*{\@gls@loadtree}{}%
  \let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
{%
  \glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
  \ifglsentrycounter
  \ifundef\c@glossaryentry
  {%
    \ifx\@gls@counterwithin\@empty
    \newcounter{glossaryentry}%

```



```

\else
  \newcounter{glossaryentry}[\@gls@counterwithin]%
\fi
\def\theHglossaryentry{\currentglossary.\theglossaryentry}%
}%
{}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glssentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
  \renewcommand*{\@gls@counterwithin}{#1}%
  \glssentrycountertrue
  \@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglssentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
  }%
  {}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{}
\glsssubentrycounterfalse
\newcommand{\glssindexingsetting}{\ifglssxindy xindy\else makeindex\fi}
\newcommand*{\@glo@default@sorttype}{standard}
\define@choicekey{glossaries.sty}{sort}{standard,def,use,none,clear}{%
  \renewcommand*{\@glo@default@sorttype}{#1}%
  \csname @gls@setupsort@#1\endcsname
}
\newcommand*{\glsprestandardsort}[3]{%
  \glsdosanitizesort
}
\newcommand*{\@glo@check@sortallowed}[1]{}
\newcommand*{\@gls@setupsort@standard}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
  \def\@gls@defsort##1##2{%
    \ifx\@glo@sort\@glsdefaultsort
      \let\@glo@sort\@glo@name
    \fi
    \let\glsdosanitizesort\@gls@sanitizesort

```

```

\glsprestandardsort{\@glo@sort}{##1}{##2}%
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{\@glo@sort}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\@gls@setupsort@standard
\newcommand*\glssortnumberfmt[1]{%
\ifnum#1<100000 0\fi
\ifnum#1<10000 0\fi
\ifnum#1<1000 0\fi
\ifnum#1<100 0\fi
\ifnum#1<10 0\fi
\number#1%
}
\newcommand*\@gls@setupsort@def{%
\def\do@glo@storeentry{\@glo@storeentry}%
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\ifcsundef{glossary@##1@sortcount}%
{\@gls@defsortcount{##1}}%
}%
\expandafter\global\expandafter
\advance\csname glossary@##1@sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@##1@sortcount\endcsname}}%
}%
\def\@gls@setsort##1{%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*\@gls@setupsort@use{%
\let\do@glo@storeentry\@gobble
\def\@gls@defsortcount##1{%
\expandafter\global
\expandafter\newcount\csname glossary@##1@sortcount\endcsname
}%
\def\@gls@defsort##1##2{%
\expandafter\gdef\csname glo@##2@sort\endcsname{}%
}%
\def\@gls@setsort##1{%
\protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
\ifx\@glo@parent\@empty
\else
\expandafter\@gls@setsort\expandafter{\@glo@parent}%
\fi
\protected@edef\@glo@type{\csname glo@##1@type\endcsname}%

```

```

\protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
\ifx\@gls@tmp\@empty
\expandafter\global\expandafter
\advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
\expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
\expandafter\glssortnumberfmt
{\csname glossary@\@glo@type @sortcount\endcsname}}%
\@glo@storeentry{##1}%
\fi
}%
\let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
\expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\@gls@setupsort@clear}{%
\def\glsindexingsetting{none}%
\def\do@glo@storeentry##1{%
\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=clear not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*{\glsdefmain}{%
\if@gls@docloaded
\newglossary[glg2]{main}{gls2}{glo2}{\glossaryname}%
\else
\newglossary{main}{gls}{glo}{\glossaryname}%
\fi
\newcommand*{\gls@tr@set@main@toctitle}{%
\translatelet{\glossarytoctitle}{Glossary}%
}%
}
\newcommand*{\glsdefaulttype}{main}
\newcommand*{\acronymtype}{\glsdefaulttype}
\@gls@declareoption{nomain}{%
\let\glsdefaulttype\relax
\renewcommand*{\glsdefmain}{}%
}

```

```

\define@boolkey{glossaries.sty}[gls]{acronym}[true]{%
  \ifglsacronym
    \renewcommand{\@gls@do@acronymsdef}{%
      \DeclareAcronymList{acronym}%
      \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
      \renewcommand*{\acronymtype}{acronym}%
      \newcommand*{\gls@tr@set@acronym@toctitle}{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%
  \ifglsacronym
    \providecommand*\printacronyms[1][1]{%
      \printglossary[type=\acronymtype,#1]%
    }%
  \fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*{\acronymtype}{acronym}%
    \newcommand*{\gls@tr@set@acronym@toctitle}{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@glsacronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@glsacronymlists\@empty
    \protected@xdef\@glsacronymlists{#1}%
  \else
    \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
  \fi
}
\newcommand*\DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\glsIfListOfAcronyms[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms[4]{%
  \def\gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%

```

```

\def\@gls@before{##1}\def\@gls@after{##2}}%
\gls@islistofacronyms,#2,#1,\@nil\end\gls@islistofacronyms
\ifx\@gls@after\@nnil
  #4%
\else
  #3%
\fi
}
\newif\if@gls@isacronymlist
\newcommand*{\gls@checkisacronymlist}[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*{\SetAcronymLists}[1]{%
  \renewcommand*{\@gls@acronymlists}{#1}%
}
\define@key{glossaries.sty}{acronymlists}{%
  \DeclareAcronymList{#1}%
}
\newcommand{\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*\glssetexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@expand@field{##1}{#1}{##2}%
}%
}
\newcommand*\glssetnoexpandfield}[1]{%
\csdef{gls@assign@#1@field}##1##2{%
\@gls@noexpand@field{##1}{#1}{##2}%
}%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*\@gls@sanitizename[]{}
\glssetnoexpandfield{name}
\newcommand*\@gls@sanitizesymbol[]{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*\@gls@sanitizesort}{%
\ifglssanitizesort
\@gls@sanitizesort
\else
\@gls@nosanitizesort
\fi

```

```

}
\newcommand*{\@@gls@sanitizesort{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@nosanitizesort}{%}
\newcommand*{\@gls@noidx@sanitizesort{%
  \ifdefvoid\@glo@sort
  {%}
  {%
    \expandafter\@@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  {%}
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*{\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent\@secondoftwo
  \let\@text@composite\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let\'@\@firstofone
  \let\'@\@firstofone
  \let\^@\@firstofone
  \let\"@\@firstofone
  \let\u@\@firstofone
  \let\t@\@firstofone
  \let\d@\@firstofone
  \let\r@\@firstofone
  \let\=@\@firstofone
  \let\.\@\@firstofone
  \let\~@\@firstofone

```

```

\let\v\@firstofone
\let\H\@firstofone
\let\c\@firstofone
\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[glS]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifglS@sanitize@description
\glSsetnoexpandfield{desc}%
\glSsetnoexpandfield{descplural}%
\else
\glSsetexpandfield{desc}%
\glSsetexpandfield{descplural}%
\fi
}
\define@boolkey[glS]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifglS@sanitize@name
\glSsetnoexpandfield{name}%
\else
\glSsetexpandfield{name}%
\fi
}
}

```



```

\define@boolkey[glS]{sanitize}{symbol}[true]{%
  \GlossariesWarning{sanitize={symbol} package option deprecated}%
  \ifglS@sanitize@symbol
    \glSsetnoexpandfield{symbol}%
    \glSsetnoexpandfield{symbolplural}%
  \else
    \glSsetexpandfield{symbol}%
    \glSsetexpandfield{symbolplural}%
  \fi
}
\define@boolkey{glossaries.sty}[glS]{sanitizesort}[true]{%
  \ifglssanitizesort
    \glSsetnoexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glssanitizesorttrue
      \glSsetnoexpandfield{sortvalue}%
    }%
  \else
    \glSsetexpandfield{sortvalue}%
    \renewcommand*{\@glS@noidx@setsanitizesort}{%
      \glssanitizesortfalse
      \glSsetexpandfield{sortvalue}%
    }%
  \fi
}
\glssanitizesorttrue
\glSsetnoexpandfield{sortvalue}%
\newcommand*{\@glS@noidx@setsanitizesort}{%
  \glssanitizesortfalse
  \glSsetexpandfield{sortvalue}%
}
\define@choicekey[glS]{sanitize}{sort}{true,false}[true]{%
  \setbool{glssanitizesort}{#1}%
  \ifglssanitizesort
    \glSsetnoexpandfield{sortvalue}%
  \else
    \glSsetexpandfield{sortvalue}%
  \fi
  \GlossariesWarning{sanitize={sort} package option
    deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
  \ifthenelse{\equal{#1}{none}}{%
    {%
      \GlossariesWarning{sanitize package option deprecated}%
      \glSsetexpandfield{name}%
      \glSsetexpandfield{symbol}%
      \glSsetexpandfield{symbolplural}%
      \glSsetexpandfield{desc}%
      \glSsetexpandfield{descplural}%
    }
  }

```

```

}%
{%
  \setkeys[glS]{sanitiz}{#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{\@language}%
\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\@glsl@labelsfile
  \immediate\openout\@glsl@labelsfile=\jobname.glslabels
  \protected@edef\@glsl@fulllist{\@glo@types}%
  \ifdefempty{\@ignored@glossaries}{}%
    {\protected@eappto\@glsl@fulllist{\@ignored@glossaries}}%
  \forallgl@sentries[\@glsl@fulllist]{\@gl@entry}%
  {%
    \ifdefempty{\@gl@entry}{}%
    {%
      \ifx\do@write@glslabelsname\@gobble
        \def\@glsl@thisentryname{%
      \else
        \letcs{\@glsl@thisentryname}{glo@\@gl@detoklabel{\@gl@entry}@name}%
        \@onelevel@sanitize\@glsl@thisentryname
      \fi
      \immediate\write\@glsl@labelsfile
        {\@gl@entry\do@write@glslabelsname{\@glsl@thisentryname}}%
    }%
  }%
  \immediate\closeout\@glsl@labelsfile
}
\newcommand*{\do@write@glslabelsname}[1]{%
  \begingroup
  \catcode'\^~I12
  \gdef\@write@glslabelsname#1{\^~I#1}
  \endgroup
  \@glsl@declareoption{writeglslabels}{\@do@write@glslabels
    \let\do@write@glslabelsname\@gobble}
  \@glsl@declareoption{writeglslabelnames}{\@do@write@glslabels
    \let\do@write@glslabelsname\@write@glslabelsname}
  \newif\ifgl@automake
  \newcommand{\glsl@automake@nr}{1}
  \newcommand{\glsl@automake@makegloss}{%
  \newcommand{\glsl@automake@option}{automake=\glsl@automake@val}
  \newcommand*{\@glsl@do@automake@err}{%
    \PackageError{glossaries}{You must use
    \string\makeglossaries\space with \glsl@automake@option}
    {%
      Either remove the \glsl@automake@option\space setting or
      add \string\makeglossaries\space to your document preamble.%
    }%
  }%
}
\define@choicekey{glossaries.sty}{automake}%
[\glsl@automake@val\glsl@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
  \ifcase\glsl@automake@nr\relax
    \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%

```

```

\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomakefalse
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\or
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\or
\def\gls@automake@nr{0}%
\glsautomaketrue
\renewcommand{\gls@automake@makegloss}{}%
\fi
\renewcommand{\gls@automake@option}{automake=\gls@automake@val}%
\ifglsautomake
\let\gls@doautomake\gls@do@automake@err
\else
\renewcommand*\{\gls@doautomake}{}%
\fi
}
\glsautomakefalse
\newcommand*\{\gls@doautomake}{%
\AtEndDocument{\gls@doautomake}
\gls@declareoption{automakegloss}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakegloss}%
\def\gls@automake@val{makegloss}%
\def\gls@automake@nr{3}%
\renewcommand{\gls@automake@makegloss}{makeglossaries}%
\let\gls@doautomake\gls@do@automake@err
}
\gls@declareoption{automakeglosslite}{%
\glsautomaketrue
\renewcommand{\gls@automake@option}{automakeglosslite}%
\def\gls@automake@val{lite}%
\def\gls@automake@nr{4}%
\renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
\let\gls@doautomake\gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
\ifglssavewrites
\renewcommand*\{\gls@writefiles}{\gls@writefiles}%
\else
\let\gls@writefiles\empty

```

```

\fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
  \ifbool{glscompatible-3.07}%
  {%
    \@gls@removedoption{compatible-3.07}%
  }%
  {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
  \ifbool{glscompatible-2.07}%
  {%
    \@gls@removedoption{compatible-2.07}%
  }%
  {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
  \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
\gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
  \ifcase\gls@kernelglossredefs@nr\relax
  \def\glossary{\@gls@override@glossary}%
  \def\makeglossary{%
    \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
    \makeglossaries
  }%
\or
  \let\glossary\gls@original@glossary

```



```

\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
  \gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
  \let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
  \newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
  \newcommand*{\printsymbols}[1][\]{\printglossary[type=symbols,##1]}%
  \newcommand*{\gls@tr@set@symbols@toctitle}{%
    \translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
  }%
}%
\@gls@declareoption{numbers}{%
  \let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
  \newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
  \newcommand*{\printnumbers}[1][\]{\printglossary[type=numbers,##1]}%
  \newcommand*{\gls@tr@set@numbers@toctitle}{%
    \translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
  }%
}%
\@gls@declareoption{index}{%
  \ifx\@gls@do@indexdef\@empty
    \let\@gls@do@indexdef\@gls@indexdef
  \fi
}
\@gls@declareoption{noglossaryindex}{%
  \let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
  \newglossary[ilg]{index}{ind}{idx}{\indexname}%
  \newcommand*{\printindex}[1][\]{\printglossary[type=index,##1]}%
  \newcommand*{\newterm}[2][\]{%
    \newglossaryentry{##2}%
    {type={index},name={##2},description={\nopostdesc},##1}}
  \let\@gls@do@indexdef\relax
}%
\define@choicekey{glossaries.sty}{mfirstuc}{%
  [\gls@mfirstuc@val\gls@mfirstuc@nr]%
  {expanded,unexpanded}%
}

```

```

{%
  \ifcase\gls@mfirstuc@nr
    % expanded
    \renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
  \or
    % unexpanded
    \renewcommand*{\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
  \fi
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
    \@expandtwoargs
    \in@ {\,\CurrentOption \,}{\,\@classoptionslist,\@curroptions,}%
    \ifin@
      \@use@ption
      \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
    \fi
  \fi
}
\ProcessOptionsX
\ifglsautomake
\ifdef\directlua{\RequirePackage{shellesc}}{}
\fi
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglsubentrycounter
    \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}[2019/11/30]
  \ifdefined\@gls@requested@languages{}%
  {%
    \@for\@gls@thislang:=\@gls@requested@languages\do{%
      \TrackIfKnownLanguage{\@gls@thislang}{}%
      {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
    }%
  }%
  \@gls@usetranslator
  \@ifpackageloaded{translator}
  {%
    \ifboolexpr
    {
      test {\ifdefstring{\trans@languages}{English}}
      and not
      test {\ifdefstring{bbl@loaded}{english}}
    }
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
    {%
      \usedictionary{glossaries-dictionary}%
      \renewcommand*{\addglossarytocaptions}[1]{%
        \ifcsundef{captions#1}{}%
        {%
          \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
          \expandafter\toks@\expandafter{\@gls@tmp
            \renewcommand*{\glossaryname}{\translate{Glossary}}}%
          }%
          \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
        }%
      }%
    }%
  }%
  {}%

```

```

\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
{}%
\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
}%
}%
\fi
\DeclareRobustCommand*\nopostdesc{}
\newcommand*{\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
}

```

```

\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
}
\@onlypremakeg\setStyleFile
\ifglxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*{\gls@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{%
\newcommand*{\glscompositor}{.}
\newcommand*{\glsSetCompositor}[1]{%
  \renewcommand*{\glscompositor}{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*{\@glsAlphacompositor}{\glscompositor}
\ifglxindy
  \newcommand*{\glsSetAlphaCompositor}[1]{%
    \renewcommand*{\@glsAlphacompositor}{#1}}
\else
  \newcommand*{\glsSetAlphaCompositor}[1]{%
    \glsnoxywarning\glsSetAlphaCompositor}
\fi
\@onlypremakeg\glsSetAlphaCompositor
\newcommand*{\gls@suffixF}{%
\newcommand*{\glsSetSuffixF}[1]{%
  \renewcommand*{\gls@suffixF}{#1}}
\@onlypremakeg\glsSetSuffixF
\newcommand*{\gls@suffixFF}{%
\newcommand*{\glsSetSuffixFF}[1]{%
  \renewcommand*{\gls@suffixFF}{#1}}
}
\ifcsundef{hyperlink}%
{%
  \newcommand*{\glsnumberformat}[1]{#1}%
}%
{%
  \newcommand*{\glsnumberformat}[1]{\glshypernumber{#1}}%
}
\newcommand{\glslocationcstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@}{\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%

```

```

    {\csgdef{@glossarypreamble@#1}{#2}}%
    {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
  }
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@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{\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@toc{#1}{#2}{%
  \ifglstoc
    \ifglsnumberline

```



```

        \addcontentsline{toc}{#2}{\protect\numberline{#1}%
\else
        \addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*\glsnxindywarning[1]{%
\GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\glsnomakeindexwarning[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\edef\xdyattributes{\string"default\string"%
\fi
\ifglsxindy
\edef\xdyattributelist{%
\fi
\ifglsxindy
\def\xdylocref{}
\fi
\newcommand*\@gls@ifinlist[4]{%
\def\@do@ifinlist##1,#1,##2\end@doifinlist{%
\def\@gls@listsuffix{##2}%
\ifx\@gls@listsuffix\@empty
#4%
\else
#3%
\fi
}%
\@do@ifinlist,#2,#1,\end@doifinlist
}
\ifglsxindy
\newcommand*\@xdycounters{\@glscounter}
\newcommand*\GlsAddXdyCounters[1]{%
\@for\@gls@ctr:=#1\do{%
\edef\@do@addcounter{%
\noexpand\@gls@ifinlist{\@gls@ctr}{\@xdycounters}{}%
{%
\noexpand\edef\noexpand\@xdycounters{\@xdycounters,%
\noexpand\@gls@ctr}%
}%
}%
\@do@addcounter
}
}
\@onlypremakeg\GlsAddXdyCounters
\else
\newcommand*\GlsAddXdyCounters[1]{%
\glsnxindywarning\GlsAddXdyAttribute

```

```

    }
\fi
\newcommand*\@disabled@glSaddxdycounters{%
    \PackageError{glossaries}{\string\GlsAddXdyCounters\space
        can't be used after \string\GlsAddXdyAttribute}{Move all
        occurrences of \string\GlsAddXdyCounters\space before the first
        instance of \string\GlsAddXdyAttribute}%
}
\ifglSxindy
\newcommand*\@glSaddxdyattribute[2]{%
    \edef\@xdyattributes{\@xdyattributes ^^J
        \string"#2#1\string"}%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\@xdylocref{\the\toks@ ^^J%
        (markup-locref
        :open \string"\glStildechar n%
        \expandafter\string\csname glsX#2X#1\endcsname
        \string" ^^J
        :close \string"\string" ^^J
        :attr \string"#2#1\string"))%
    \expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
        \setentrycounter[##1][##2]\csname #1\endcsname{##2}%
    }%
}
\newcommand*\GlsAddXdyAttribute[1]{%
    \ifx\@xdyattributelist\@empty
        \edef\@xdyattributelist{#1}%
    \else
        \edef\@xdyattributelist{\@xdyattributelist,#1}%
    \fi
    \edef\@xdyattributes{\@xdyattributes ^^J
        \string"#1\string"}%
    \@for\@this@counter:=\@xdycounters\do{%
        \protected@edef\glS@do@addxdyattribute{%
            \noexpand\@glSaddxdyattribute{#1}{\@this@counter}%
        }
        \glS@do@addxdyattribute
    }%
    \let\GlsAddXdyCounters\@disabled@glSaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
    \glSnoxindywarning\GlsAddXdyAttribute}
\fi
\ifglSxindy
\newcommand*{\@glS@addpredefinedattributes}{%
    \GlsAddXdyAttribute{glSnumberformat}
    \GlsAddXdyAttribute{textrm}
    \GlsAddXdyAttribute{textsf}

```

```

\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{hyperrm}
\GlsAddXdyAttribute{hypersf}
\GlsAddXdyAttribute{hypertt}
\GlsAddXdyAttribute{hyperbf}
\GlsAddXdyAttribute{hypermd}
\GlsAddXdyAttribute{hyperit}
\GlsAddXdyAttribute{hyperup}
\GlsAddXdyAttribute{hypersl}
\GlsAddXdyAttribute{hypersc}
\GlsAddXdyAttribute{hyperemph}
\GlsAddXdyAttribute{glsignore}
}
\else
\let\@gls@addpredefinedattributes\relax
\fi
\def\@xdyuseralphabets{
\ifglxsindy
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ^~J
(define-alphabet "#1" (#2))}}
\else
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxsindy
\newcommand*{\@gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%
arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@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 "\glsclosebrace"
    #1
    :sep "\glsclosebrace\glsclosebrace" #3
    :sep "\glsclosebrace"))
}%
\fi
\edef\@xdyuserlocationnames{%
  \@xdyuserlocationnames^^J\space\space\space
  \string"#2\string"%
}
\@onlypremakeg\GlsAddXdyLocation
\else
  \newcommand*\GlsAddXdyLocation}[2]{%
    \glsnxindywarning\GlsAddXdyLocation}
\fi
\ifglxsindy
\def\@xdylocationclassorder{^^J\space\space\space
  \string"roman-page-numbers\string"^^J\space\space\space
  \string"arabic-page-numbers\string"^^J\space\space\space
  \string"arabic-section-numbers\string"^^J\space\space\space
  \string"alpha-page-numbers\string"^^J\space\space\space
  \string"Roman-page-numbers\string"^^J\space\space\space
  \string"Alpha-page-numbers\string"^^J\space\space\space
  \string"Appendix-page-numbers\string"
  \@xdyuserlocationnames^^J\space\space\space
  \string"see\string"
}
\fi
\ifglxsindy
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \def\@xdylocationclassorder{#1}}
\else
  \newcommand*\GlsSetXdyLocationClassOrder[1]{%
    \glsnxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxsindy
  \def\@xdysortrules{}
\fi
\ifglxsindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))}%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnxindywarning\GlsAddSortRule}

```

```

\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnxindywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnxindywarning\GlsSetXdyStyles}
\fi
\def\@xdy@language#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyLanguage[2][\glsdefaulttype]{%
    \ifglossaryexists{#1}{%
      \expandafter\def\csname @xdy@#1@language\endcsname{#2}%
    }{%
      \PackageError{glossaries}{Can't set language type for
        glossary type '#1' --- no such glossary}{%
        You have specified a glossary type that doesn't exist}}
\else
  \newcommand*\GlsSetXdyLanguage[2][]{%
    \glsnxindywarning\GlsSetXdyLanguage}
\fi
\def\@gls@codepage#1#2{}
\ifglxindy
  \newcommand*\GlsSetXdyCodePage[1]{%
    \renewcommand*\@gls@codepage{#1}%
  }
  \AtBeginDocument{%
    \ifx\@gls@codepage\@empty
      \ifpackageloaded{fontspec}{\def\@gls@codepage{utf8}}{}%
    \fi
  }
\else
  \newcommand*\GlsSetXdyCodePage[1]{%
    \glsnxindywarning\GlsSetXdyCodePage}
\fi
\ifglxindy
  \ifglx@xindy@glsnumbers
    \def\@xdylettergroups{(define-letter-group
      \string"glsnumbers\string"^^J\space\space\space
      :prefixes (\string"0\string" \string"1\string"

```

```

\string"2\string" \string"3\string" \string"4\string"
\string"5\string" \string"6\string" \string"7\string"
\string"8\string" \string"9\string")^^J\space\space\space
\@xdynumbergrouporder)}}
\else
\def\@xdylettergroups{}
\fi
\fi
\newcommand*\GlsAddLetterGroup[2]{%
\expandafter\toks@\expandafter{\@xdylettergroups}%
\protected@edef\@xdylettergroups{\the\toks@^^J%
(define-letter-group \string"#1\string"^^J\space\space\space#2)}%
}%
\newcommand*\forallglossaries[3][\@glo@types]{%
\@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
}

\newcommand*\forallacronyms[2]{%
\@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forallglsentries[3][\glsdefaulttype]{%
\protected@edef\@glo@list{\csname glolist@#1\endcsname}%
\@for#2:=\@glo@list\do
{%
\ifdefempty{#2}{\fi}%
}%
}
\newcommand*\forallglsentries[3][\@glo@types]{%
\expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
{%
\forallglsentries[\@this@glo@]{#2}{#3}%
}%
}
\newcommand{\ifglossaryexists}{%
\@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
\ifcsundef{\@glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
\ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*\glsdetoklabel[1]{#1}
\newcommand*\ifglsentryexists[3]{%
\ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*\ifglsused[3]{%
\ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand*\glsdoifexists[2]{%

```

```

\ifglentryexists{#1}{#2}{%
  \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
    has not been defined}{You need to define a glossary entry before you
    can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{}}{#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}
\newcommand{\glsdoifexistsordo}[3]{%
  \ifglentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
  #3%
}%
}
\newcommand{\doifglossarynoexistsordo}[3]{%
  \ifglossaryexists*{#1}%
  {%
    \PackageError{glossaries}{Glossary type ‘#1’ already exists}{}%
    #3%
  }%
  {#2}%
}
\newrobustcmd{\ifglshaschildren}[3]{%
  \glsdoifexists{#1}%
  {%
    \def\do@glshaschildren{#3}%
    \protected@edef\@gls@thislabel{\glsdetoklabel{#1}}%
    \expandafter\for@glentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs@glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
        {%
          \def\do@glshaschildren{#2}%
          \@endfortrue
        }%
      }%
    }%
  }%
  \do@glshaschildren
}

```



```

    }%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \ifcsemtty{glo@glstetoklabel{#1}@parent}{#3}{#2}%
  }%
}
\newcommand*{\ifglshasdesc}[3]{%
  \ifcsemtty{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}%
  }%
}
\newcommand{\ifglstfieldvoid}[4]{\ifcsvoid{glo@glstetoklabel{#2}@#1}{#3}{#4}}
\newrobustcmd*{\ifglshasfield}[4]{%

```

```

\glsdoifexists{#2}%
{%
  \letcs{\@glo@thisvalue}{glo@glsetoklabel{#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@glsetoklabel{#2}@ \@gls@thisfield}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
    {%
      \ifdefequal\@glo@thisvalue\@gls@default@value
      {%
        #4%
      }%
      {%
        \let\glscurrentfieldvalue\@glo@thisvalue
        #3%
      }%
    }%
  }%
}
{%
  \GlossariesWarning{Unknown entry field '#1'}%
  #4%
}
}%
}
\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary{%

```

```

\protected@write\@auxout{}\string\providecommand\string\@newglossary[4]{}%
\let\@gls@provide@newglossary\relax
}
\newcommand*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
\protected@edef\glo@tmp{#1}%
\ifx\glo@tmp\@gls@defaulttype
\GlossariesWarning{\string\defglsentryfmt\space attempting to set
a display style for \string\glsdefaulttype\space but there
doesn't seem to be a default glossary. If you have used
'nomain', you need to define an alternative glossary}%
\else
\csgdef{gls@\glo@tmp @entryfmt}{#2}%
\fi
}
\newcommand*{\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtexext}[1]{%
\ifboolexpr{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
\ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaulttype
{%
\gdef\glsdefaulttype{#2}%
}{}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtexext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtexext\csname @glotype@#2@out\endcsname

```

```

\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\auxout{}\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
  \defglsentryfmt[#2]{\glsentryfmt}%
}%
{}%
\@gls@defsortcount{#2}%
\@ifnextchar[{\@gls@setcounter{#2}}%
  {\@gls@setcounter{#2}[\glscounter]}%
}%
{%
  \gls@gobbleopt
}%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxindy
    \GlsAddXdyCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{, #1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
}

```

```

\ifdefempty\@gls@nohyperlist
{%
  \renewcommand*{\@gls@nohyperlist}{#1}%
}%
{%
  \protected@eappto\@gls@nohyperlist{, #1}%
}%
}
\newcommand*{\@ignored@glossaries}{}
\newcommand*{\ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
    {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%
\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*{\@gls@default@value}{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%

```

```

    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@glo@counter{#1}%
  }%
}
\define@key{glossentry}{see}{%
  \gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
  \renewcommand*{\gls@xr@key}{#1}%
  \gls@checkseeallowed
  \def#2{#3}%
  \@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
  \@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
  \GlossariesWarning
  {‘\gls@xr@key’ key doesn’t have any effect when used in the document
  environment. Move the definition to the preamble
  after \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
  \def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
[\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
  \ifcase\gls@nonumberlist@nr\relax
    \def\@glo@prefix{\glsnonextpages}%
    \@gls@savenonumberlist{true}%
  \else
    \def\@glo@prefix{\glsnextpages}%
    \@gls@savenonumberlist{false}%
  \fi
}
\newcommand*{\gls@noidx@nonumberlist}[1]{%
  \cslet\glo@\@glo@label @prenumberlist\@glo@prefix
}%
\newcommand{\glsnoidxprenumberlist}[1]{%
  \csuse{glo@\glsdetoklabel{#1}@prenumberlist}}
\newcommand*{\@gls@savenonumberlist}[1]{%
}
\newcommand*{\@gls@initnonumberlist}{%
}

```

```

\newcommand*{\@gls@storenonumberlist}[1]{%
\newcommand*{\@gls@enablesavenonumberlist}{%
  \renewcommand*{\@gls@initnonumberlist}{%
    \undef\@glo@nonumberlist
  }%
  \renewcommand*{\@gls@savenonumberlist}[1]{%
    \def\@glo@nonumberlist{##1}%
  }%
  \renewcommand*{\@gls@storenonumberlist}[1]{%
    \ifdef\@glo@nonumberlist
    {%
      \cslet{glo@glsdetoklabel{##1}@nonumberlist}{\@glo@nonumberlist}%
    }%
    {}%
  }%
  \appto\@gls@keymap{,{nonumberlist}{nonumberlist}}%
}
\define@key{glossentry}{user1}{%
  \def\@glo@useri{#1}%
}
\define@key{glossentry}{user2}{%
  \def\@glo@userii{#1}%
}
\define@key{glossentry}{user3}{%
  \def\@glo@useriii{#1}%
}
\define@key{glossentry}{user4}{%
  \def\@glo@useriv{#1}%
}
\define@key{glossentry}{user5}{%
  \def\@glo@userv{#1}%
}
\define@key{glossentry}{user6}{%
  \def\@glo@uservi{#1}%
}
\define@key{glossentry}{short}{%
  \def\@glo@short{#1}%
}
\define@key{glossentry}{shortplural}{%
  \def\@glo@shortpl{#1}%
}
\define@key{glossentry}{long}{%
  \def\@glo@long{#1}%
}
\define@key{glossentry}{longplural}{%
  \def\@glo@longpl{#1}%
}
\newcommand*{\@gls@name}{%
  \PackageError{glossaries}{name key required in
  \string\newglossaryentry\space for entry '\@glo@label'}{You

```

```

        haven't specified the entry name}}
\newcommand*{\@glsnodelsc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '\@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'.
      Maybe makeglossaries + rerun required}%
  \else
    \PackageError{glossaries}%
    {Package option 'savenumberlist=true' required}%
    {%
      You must use the 'savenumberlist' package option
      to reference location lists.%
    }%
  \fi
}
\newcommand*{\@gls@defaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
  \expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
  \ifcsdef{gls@assign@#3@field}
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
    }%
    {%
      \csuse{gls@assign@#3@field}{#2}{#4}%
    }%
  }%
  {%
    \ifdefequal{#4}{\@gls@default@value}%
    {%
      \edef\@gls@value{\expandonce{#1}}%
      \@gls@noexpand@field{#2}{#3}{\@gls@value}%
    }%
    {%

```



```

        \@gls@noexpand@field{#2}{#3}{#4}%
    }%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
\protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\edef\@gls@value{\expandonce{#1}}%
\csuse{gls@assign@#3@field}{#2}{\@gls@value}%
}%
{%
\expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
{%
\csuse{gls@assign@#3@field}{#2}{#4}%
}%
}%
}%
{%
\ifdefequal{#4}{\@gls@default@value}%
{%
\@gls@expand@field{#2}{#3}{#1}%
}%
{%
\@gls@expand@field{#2}{#3}{#4}%
}%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
\def\@gls@tmp{#1}%
\ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*\{gls@expandfields}{%
\let\gls@assign@field\@gls@expand@fields
}
\newcommand*\{gls@noexpandfields}{%
\let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
\glsdoifnoexists{#1}%

```

```

    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \newcommand*{\gls@defdocnewglossaryentry}{%
    \let\newglossaryentry\new@glossaryentry
  }
  \newrobustcmd{\provideglossaryentry}[2]{%
    \ifglentryexists{#1}%
    {%
      \gls@defglossaryentry{#1}{#2}%
    }%
  }
  \@onlypreamble{\provideglossaryentry}
  \newcommand*{\gls@gldefs@labellist}{}
  \newcommand*{\gls@gldefs@addlabel}[1]{\listxadd{\gls@gldefs@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
    }%
    {%
      \ifglentryexists{#1}%
      {%
        \xifinlist{#1}{\gls@docdef@labellist}%
        {%
          \GlossariesWarning{Entry ‘#1’ has already been defined in the
            document environment (ignoring new definition)}%
        }%
        {%
          \xifinlist{#1}{\gls@gldefs@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\}\glpercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%
  {user1}{useri},%
  {user2}{userii},%
  {user3}{useriii},%
  {user4}{useriv},%
  {user5}{userv},%
  {user6}{uservi},%
  {long}{long},%
  {longplural}{longpl},%
  {short}{short},%
  {shortplural}{shortpl},%
  {counter}{counter},%
  {parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
}%
{}%

```

```

\@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
}%
{%
\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\@sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@@gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{, {#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{\@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@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}{\glssupercase{#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{\glsfieldxddef}[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{\glsfieldgdgdef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%
\ifcsdef{glo@\@glo@label @#2}%
{%
\expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
}%
{%
\PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}
\newrobustcmd{\glsfieldddef}[3]{%
\glsdoifexists{#1}%
{%
\protected@edef\@glo@label{\glsdetoklabel{#1}}%

```

```

\ifcsdef{glo@\@glo@label @#2}%
{%
  \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
}%
{%
  \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
}%
}%
}
\newrobustcmd{\glsfieldfetch}[3]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \letcs#3{glo@\@glo@label @#2}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
  \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
  \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
  \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
\glsdoifexists{#1}%
{%
  \protected@edef\@glo@label{\glsdetoklabel{#1}}%
  \ifcsdef{glo@\@glo@label @#2}%
  {%
    \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
  }%
  {%
    \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
  }%
}

```



```

    }%
  }%
}
\newcommand{\ifglsfieldcseq}[5]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\glo@label @#2}%
    {%
      \ifcsstrequal{glo@\glo@label @#2}{#3}{#4}{#5}%
    }%
    {%
      \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newcommand*{\glswritedefhook}{%
\newcommand*{\gls@assign@desc}[1]{%
  \gls@assign@field{#1}{desc}{\@glo@desc}%
  \gls@assign@field{\@glo@desc}{#1}{descplural}{\@glo@descplural}%
}
\newcommand{\longnewglossaryentry}[3]{%
  \glsdoifnoexists{#1}%
  {%
    \bgroup
    \let\@org@newglossaryentryprehook\@newglossaryentryprehook
    \long\def\@newglossaryentryprehook{%
      \long\def\@glo@desc{#3\leavevmode\unskip\nopostdesc}%
      \@org@newglossaryentryprehook
    }%
    \renewcommand*{\gls@assign@desc}[1]{%
      \global\cslet{glo@\glsdetoklabel{#1}@desc}{\@glo@desc}%
      \global\cslet{glo@\glsdetoklabel{#1}@descplural}{\@glo@desc}%
    }
    \gls@defglossaryentry{#1}{#2}%
  \egroup
}
}
\@onlypreamble{\longnewglossaryentry}
\newcommand{\longprovideglossaryentry}[3]{%
  \ifglsentryexists{#1}{%
    {\longnewglossaryentry{#1}{#2}{#3}}%
  }
}
\@onlypreamble{\longprovideglossaryentry}
\newcommand{\gls@defglossaryentry}[2]{%
  \let\GlsSetQuote\gls@nosetquote
  \protected@edef\glo@label{\glsdetoklabel{#1}}%
  \let\glslabel\@glo@label
  \let\@glo@name\@gls@name
  \let\@glo@desc\@gls@desc

```

```

\let\@glo@descplural\@gls@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\@glo@list@\csname glo@list@\@glo@type\endcsname}%
\expandafter\protected@xdef\csname glo@list@\@glo@type\endcsname{%
  \@glo@list@\@glo@label},}%
}%
\gls@level=0\relax
\ifx\@glo@parent\@empty
  \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
\else
  \ifdefequal\@glo@label\@glo@parent%
  {%
    \PackageError{glossaries}{Entry ‘\@glo@label’ can’t be its own parent}{}%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
  {%
    \ifglsentryexists{\@glo@parent}%
    {%
      \expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
        \@glo@parent}%
      \gls@level=\csname glo@\@glo@parent @level\endcsname\relax
      \advance\gls@level by 1\relax
      \ifx\@glo@name\@glsnoname
        \expandafter\let\expandafter\@glo@name
        \csname glo@\@glo@parent @name\endcsname
      \ifx\@glo@plural\@gls@default@value
        \expandafter\let\expandafter\@glo@plural
        \csname glo@\@glo@parent @plural\endcsname
      \fi
    \fi
  }%
  {%
    \PackageError{glossaries}%
    {%
      Invalid parent ‘\@glo@parent’
      for entry ‘\@glo@label’ - parent doesn’t exist%
    }%
    {%
      Parent entries must be defined before their children%
    }%
    \def\@glo@parent{}%
    \expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
  }%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%

```

```

\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname\glspluralsuffix}%
  {\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
  {\csname glo@\glo@label @text\endcsname}%
  {\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @plural\endcsname}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\else
  \expandafter\gls@assign@field\expandafter
    {\csname glo@\glo@label @first\endcsname\glspluralsuffix}%
    {\@glo@label}{firstpl}{\@glo@firstplural}%
\fi
\ifcsundef{@glotype@\@glo@type @counter}%
{%
  \def\@glo@defaultcounter{\glscounter}%
}%
{%
  \letcs\@glo@defaultcounter{@glotype@\@glo@type @counter}%
}%
\gls@assign@field{\@glo@defaultcounter}{\@glo@label}{counter}{\@glo@counter}%
\gls@assign@field{}{\@glo@label}{useri}{\@glo@useri}%
\gls@assign@field{}{\@glo@label}{userii}{\@glo@userii}%
\gls@assign@field{}{\@glo@label}{useriii}{\@glo@useriii}%
\gls@assign@field{}{\@glo@label}{useriv}{\@glo@useriv}%
\gls@assign@field{}{\@glo@label}{userv}{\@glo@userv}%
\gls@assign@field{}{\@glo@label}{uservi}{\@glo@uservi}%
\gls@assign@field{}{\@glo@label}{short}{\@glo@short}%
\gls@assign@field{}{\@glo@label}{shortpl}{\@glo@shortpl}%
\gls@assign@field{}{\@glo@label}{long}{\@glo@long}%
\gls@assign@field{}{\@glo@label}{longpl}{\@glo@longpl}%
\ifx\@glo@name\@gls@noname
  \@gls@noname
  \let\@glo@name\@gls@default@value
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@\glo@label @numberlist}%
{%
  \csxdef{glo@\glo@label @numberlist}{%
    \noexpand\@gls@missingnumberlist{\@glo@label}}%
}%
}%
\@gls@storenonumberlist{\@glo@label}%
\def\@glo@@desc{\@glo@first}%
\ifx\@glo@desc\@glo@@desc
  \let\@glo@desc\@glo@first
\fi
\ifx\@glo@desc\@gls@nodesc

```

```

\@glsnodesc
\let\@glodesc\@gls@default@value
\fi
\gls@assign@desc{\@glo@label}%
\@gls@defsort{\@glo@type}{\@glo@label}%
\def\@glo@symbol{\@glo@text}%
\ifx\@glo@symbol\@glo@symbol
\let\@glo@symbol\@glo@text
\fi
\gls@assign@field{\relax}{\@glo@label}{symbol}{\@glo@symbol}%
\expandafter
\gls@assign@field\expandafter
{ \csname glo@\@glo@label @symbol\endcsname }
{ \@glo@label }{symbolplural}{ \@glo@symbolplural }%
\expandafter\protected@xdef\csname glo@\@glo@label @flagfalse\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iffalse
}%
\expandafter\protected@xdef\csname glo@\@glo@label @flagtrue\endcsname{%
\noexpand\global
\noexpand\let\expandafter\noexpand
\csname ifglo@\@glo@label @flag\endcsname\noexpand\iftrue
}%
\csname glo@\@glo@label @flagfalse\endcsname
\@glo@autosee
\ifignoredglossary\@glo@type
{%
\csdef{glo@\@glo@label @index}{}%
}
{%
\do@glo@storeentry{\@glo@label}%
}%
\@newglossaryentry@defcounters
\@newglossaryentryposthook
}
\newcommand*{\@glo@autosee}{%
\ifdefined\@glo@see{%
{%
\protected@edef\@do@glssee{%
\noexpand\@gls@fixbraces\noexpand\@glo@list\@glo@see\noexpand\@nil
\noexpand\expandafter\noexpand\@glssee\noexpand\@glo@list{\@glo@label}}%
\@do@glssee
}%
\@glo@autoseehook
}%
\newcommand*{\@glo@autoseehook}{}
\newcommand*{\@newglossaryentryprehook}{}
\newcommand*{\@newglossaryentryposthook}{}
\newcommand*{\@newglossaryentry@defcounters}{}

```

```

\newcommand*{\glsmoveentry}[2]{%
  \protected@edef\@glo@thislabel{\glsdetoklabel{#1}}%
  \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}%
  \ifglxindy
    \ifx\@glo@parent\@empty
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
      }%
    \else
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \csname glo@\@glo@parent @index\endcsname
        (\string"\@glo@sort\string" %
        \string"\@glo@prefix\@glossarysubentryfield
        {\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
      }%
    \fi
  \else
    \ifx\@glo@parent\@empty
      \@onelevel@sanitize\@glo@prefix
      \expandafter\protected@xdef\csname glo@#1@index\endcsname{%
        \@glo@sort\@gls@actualchar\@glo@prefix

```

```

        \@glossaryentryfield{\@glo@esclabel}%
    }%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*{\@gls@if@measuring}[2]{%
\ifmeasuring@
#1%
\else
#2%
\fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand{\@gls@if@measuring}[2]{#2}
\AtBeginDocument{%
\@ifpackageloaded{amsmath}%
{\let\gls@if@measuring\@gls@if@measuring}%
{}%
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\@glsifmeasuring{}\@#1}
\newcommand*\@glsifmeasuring{\@gls@if@measuring}
\newcommand*\@gls@local@stepcounter[1]{%
\ifcsdef{c@#1}%
{%
\epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
\expandafter\advance\csname c@#1\endcsname by \@ne
}%
{}%
}
\newcommand{\@gls@start@measuring}{%
\ifx\glsifmeasuring\@firstoftwo
\else
\let\gls@org@target\glstarget
\let\gls@org@link\@glslink
\def\gls@measuring@reset{}%
\fi
\let\glsifmeasuring\@firstoftwo
\let\refstepcounter\gls@local@stepcounter
\let\label\@gobble
\let\glstarget\@secondoftwo

```

```

\let\@glslink\@secondoftwo
}
\newcommand{\gls@stop@measuring}{%
\renewcommand*{\gls@ifmeasuring}{\gls@if@measuring}%
\let\refstepcounter\gls@org@refstepcounter
\let\label\gls@org@label
\let\glstarget\gls@org@target
\let\@glslink\gls@org@link
\gls@measuring@reset
\def\gls@measuring@reset{%
}
\newcommand{\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*\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}%
  {%
    \@glsunset{#1}%
  }%
}%
}
\newcommand*{\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \@glslocalunset{#1}%
    }%
  }%
}
\newcommand*{\@glslocalunset}{\@glslocalunset}
\newcommand*{\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}

```

```

}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*{\@newglossaryentry@defcounters}{%
  \csdef{glo@\@glo@label @currcount}{0}%
  \csdef{glo@\@glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*{\glsenableentrycount}{%
  \let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
  \renewcommand*{\gls@defdocnewglossaryentry}{%
    \renewcommand*{\newglossaryentry}[2]{%
      \PackageError{glossaries}{\string\newglossaryentry\space
        may only be used in the preamble when entry counting has
        been activated}{If you use \string\glsenableentrycount\space
        you must place all entry definitions in the preamble not in
        the document environment}%
    }%
  }%
}
\newcommand*{\glsentrycurrcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@currcount}%
  {0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
  \ifcsundef{glo@\glsdetoklabel{##1}@prevcount}%
  {0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
  \@glsunset{##1}%
  \@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
  \@glslocalunset{##1}%
  \@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
  \@glsreset{##1}%
  \ifglsresetcurrcount\csgdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*{\@glslocalreset}[1]{%
  \@glslocalreset{##1}%
  \ifglsresetcurrcount\csdef{glo@\glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgls@##1##2[##3]{%
  \ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@gl@{##1}{##2}[##3]%
    \fi
}%
\def\@cgl@{##1}{##2}[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@gl@{##1}{##2}[##3]%
    \fi
}%
\def\@Gls@{##1}{##2}[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Gls@{##1}{##2}[##3]%
    \fi
}%
\def\@Glspl@{##1}{##2}[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \@Glspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gl@write@entrycounts}%
\renewcommand*{\@gl@entry@count}[2]{%
    \csgdef{glo@\glsetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gl@increment@currcount}[1]{%
    \csxdef{glo@\glsetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gl@local@increment@currcount}[1]{%
    \csedef{glo@\glsetoklabel{##1}@currcount}{%
        \number\numexpr\glsetcurrcount{##1}+1}%
}
\newcommand*{\@gl@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*\@gl@entry@count}[2]{}}%
\forallglsetries{\@glset}{%
    \ifglsetused{\@glset}%

```

```

        {\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}
\glsmfuaddmap{\cgl}{\cGls}
\newcommand*{\@cGls}[2][{}]{%
    \new@ifnextchar[{\@cGls@{\#1}{\#2}}{\@cGls@{\#1}{\#2}[]}%
}
\def\@cGls@#1#2[#3]{%
    \GlossariesWarning{\string\cGls\space is defaulting to
        \string\Gls\space since you haven't enabled entry counting}%
    \@Gls@{\#1}{\#2}[#3]%
}
\newcommand*{\cGlsformat}[2]{%
    \ifglshaslong{\#1}{\Glsentrylong{\#1}}{\Glsentryfirst{\#1}}#2%
}
\newrobustcmd*{\cglspl}{\@gls@hyp@opt\@cglspl}
\newcommand*{\@cglspl}[2][{}]{%
    \new@ifnextchar[{\@cglspl@{\#1}{\#2}}{\@cglspl@{\#1}{\#2}[]}%
}
\def\@cglspl@#1#2[#3]{%
    \GlossariesWarning{\string\cglspl\space is defaulting to
        \string\glspl\space since you haven't enabled entry counting}%
    \@glspl@{\#1}{\#2}[#3]%
}
\newcommand*{\cglsplformat}[2]{%
    \ifglshaslong{\#1}{\glsentrylongpl{\#1}}{\glsentryfirstplural{\#1}}#2%
}
\newrobustcmd*{\cGlspl}{\@gls@hyp@opt\@cGlspl}
\glsmfuaddmap{\cglspl}{\cGlspl}
\newcommand*{\@cGlspl}[2][{}]{%
    \new@ifnextchar[{\@cGlspl@{\#1}{\#2}}{\@cGlspl@{\#1}{\#2}[]}%
}
\def\@cGlspl@#1#2[#3]{%
    \GlossariesWarning{\string\cGlspl\space is defaulting to

```

```

\string\Glspl\space since you haven't enabled entry counting}%
\@Glspl@{#1}{#2}{#3}%
}
\newcommand*{\cGlsplformat}[2]{%
  \ifglshaslong{#1}{\Glsentrylongpl{#1}}{\Glsentryfirstplural{#1}}#2%
}
\newcommand*{\loadglsentries}[2][\@gls@default]{%
  \let\@gls@default\glsdefaulttype
  \def\glsdefaulttype{#1}\input{#2}%
  \let\glsdefaulttype\@gls@default
}
\@onlypreamble{\loadglsentries}
\newcommand*{\glstextformat}[1]{#1}
\newcommand*{\glsentryfmt}{\glsgenentryfmt}
\newcommand*{\glsgenentryfmt}{%
  \ifdefempty\glscustomtext
  {%
    \glsifplural
    {%
      \glscapscase
      {%
        \ifglsused\glslabel
        {%
          \glsentryplural{\glslabel}\glsinsert
        }%
        {%
          \glsentryfirstplural{\glslabel}\glsinsert
        }%
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \Glsentryplural{\glslabel}\glsinsert
      }%
      {%
        \Glsentryfirstplural{\glslabel}\glsinsert
      }%
    }%
    {%
      \ifglsused\glslabel
      {%
        \glsuppercase{\glsentryplural{\glslabel}\glsinsert}%
      }%
      {%
        \glsuppercase{\glsentryfirstplural{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase

```

```

{%
  \ifglused\glslabel
  {%
    \glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \glsupercase{\glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \glsupercase{\glsentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \glsupercase
            {\acronymfont{\glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
  }%
}

```

```

    }%
}%
{%
  \glscapscase
  {%
    \acronymfont{\glentryshort{\glslabel}}\glsinsert
  }%
  {%
    \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
  }%
  {%
    \glsupercase
    {\acronymfont{\glentryshort{\glslabel}}\glsinsert}%
  }%
}%
}%
{%
  \glsifplural
  {%
    \glscapscase
    {%
      \genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \Genplacrfullformat{\glslabel}{\glsinsert}%
    }%
    {%
      \glsupercase
      {\genplacrfullformat{\glslabel}{\glsinsert}}%
    }%
  }%
}%
{%
  \glscapscase
  {%
    \genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \Genacrfullformat{\glslabel}{\glsinsert}%
  }%
  {%
    \glsupercase
    {\genacrfullformat{\glslabel}{\glsinsert}}%
  }%
}%
}%
}%
{%
  \glscustomtext
}%
}

```

```

\newcommand*{\genacrfullformat}[2]{%
  \glstrylong{#1}#2\space
  (\protect\firstacronymfont{\glstryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
  \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\newcommand*{\genplacrfullformat}[2]{%
  \glstrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glstryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
}%
\def\@gls@counter{#1}%
}%
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*{\@gls@glslink@hyper@update@hook}{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef\hyperlink{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset\glslocalunset
  \else
    \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*{\glslinkvar}[3]{#1}

```



```

\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree
  \@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@thirdofthree
  \@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
  \@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3][]{%
  \glsdoifexistsordo{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\relax
    \@gls@link[#1]{#2}{#3}%
  }{%
    \glstextformat{#3}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
  \@gls@hyp@opt\@Gls@@link
}
\newcommand*{\@Gls@@link}[3][]{%
  \@gls@@link[#1]{#2}{\glssentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
  \ifglsused{\glslabel}%
  {%
  }%
  {%
    \gls@checkisacronymlist\glstype
    \ifglshyperfirst
      \if@glsisacronymlist
        \ifglsacrfootnote
          \KV@glslink@hyperfalse
        \fi
      \fi
    \else
      \KV@glslink@hyperfalse
    \fi
  }%
  \glslinkcheckfirsthyperhook
}

```

```

}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhypon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@gl:disablehyperinlist}{%
  \expandafter\DTLifinlist\expandafter{\gls@type}{\@gls@nohyperlist}%
  {\KV@glslink@hyperfalse}}%
}
\newcommand*{\@gls@setdefault@glslink@opts}{}
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{\glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\gls@type{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper\ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@gl:disablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@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
  \@gls@restore@glslocal
}
\newcommand*{\glsencapwrcontent}[1]{#1}
\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@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}%
\ifglsclocations
\@for\@gls@tmp:=\gls@protected@pagefmts\do
{%
\edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\expandonce\@gls@tmp}%
\@onelevel@sanitize\@gls@sanitized@tmp
\edef\gls@dosubst{%
\noexpand\DTLsubstituteall\noexpand\gls@xdystring
{\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
}%
\gls@dosubst
}%
\fi
\let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
\ifglsxindy
\@gls@escbsdq{#1}%
\else
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkquote#1\@nil""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescquote#1\@nil\""\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescactual#1\@nil\?'\?'\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%

```

```

\expandafter\@gls@checkactual#1\@nil??\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkbar#1\@nil||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkescbar#1\@nil|||\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checklevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\expandafter\@gls@checkesclevel#1\@nil!!\null
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\def\@gls@checkedmkidx{}%
\edef\@do@gls@checkesc{%
  \noexpand\@gls@checkesc\expandonce{#1}\noexpand\@nil
  \@backslashchar\@backslashchar\noexpand\null
}%
\@do@gls@checkesc
\expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
\fi
}
\def\@gls@updatechecked#1\@nil#2{\def#2{#1}}
\newtoks\@gls@tmpb
\def\@gls@checkquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
  \toks@={#1}%
  \ifx\null#2\null
    \ifx\null#3\null
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
      \def\@@gls@checkquote{\relax}%
    \else
      \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
        \@gls@quotechar\@gls@quotechar\@gls@quotechar\@gls@quotechar}%
      \def\@@gls@checkquote{\@gls@checkquote#3\null}%
    \fi
  \else
    \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
      \@gls@quotechar\@gls@quotechar}%
    \ifx\null#3\null
      \def\@@gls@checkquote{\@gls@checkquote#2""\null}%
    \else
      \def\@@gls@checkquote{\@gls@checkquote#2"#3\null}%
    \fi
  \fi
  \@@gls@checkquote
}
\def\@gls@checkescquote#1"#2"#3\null{%
  \@gls@tmpb=\expandafter{\@gls@checkedmkidx}%

```

```

\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar
\@gls@quotechar\string"\@gls@quotechar}%
\def\@gls@checkescquote{\@gls@checkescquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@quotechar}%
\ifx\null#3\null
\def\@gls@checkescquote{\@gls@checkescquote#2\""\null}%
\else
\def\@gls@checkescquote{\@gls@checkescquote#2\">#3\null}%
\fi
\fi
\@gls@checkescquote
}
\def\@gls@checkescactual#1\?#2\?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar
\@gls@quotechar\string"\@gls@actualchar}%
\def\@gls@checkescactual{\@gls@checkescactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkescactual{\@gls@checkescactual#2\?\?\null}%
\else
\def\@gls@checkescactual{\@gls@checkescactual#2\?#3\null}%
\fi
\fi
\@gls@checkescactual
}
\def\@gls@checkescbar#1\|#2\|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null

```

```

\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkescbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar
\@gls@quotechar\string"\@gls@encapchar}%
\def\@gls@checkescbar{\@gls@checkescbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkescbar{\@gls@checkescbar#2\|\|\null}%
\else
\def\@gls@checkescbar{\@gls@checkescbar#2\|#3\null}%
\fi
\fi
\@gls@checkescbar
}
\def\@gls@checkesclevel#1\!#2\!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesclevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar
\@gls@quotechar\string"\@gls@levelchar}%
\def\@gls@checkesclevel{\@gls@checkesclevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\string"\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!\!\null}%
\else
\def\@gls@checkesclevel{\@gls@checkesclevel#2\!#3\null}%
\fi
\fi
\@gls@checkesclevel
}
\def\@gls@checkbar#1|#2|#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%

```

```

\def\@gls@checkbar{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar\@gls@quotechar\@gls@encapchar}%
\def\@gls@checkbar{\@gls@checkbar#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@encapchar}%
\ifx\null#3\null
\def\@gls@checkbar{\@gls@checkbar#2||\null}%
\else
\def\@gls@checkbar{\@gls@checkbar#2|#3\null}%
\fi
\fi
\@gls@checkbar
}
\def\@gls@checklevel#1!#2!#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checklevel{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar\@gls@quotechar\@gls@levelchar}%
\def\@gls@checklevel{\@gls@checklevel#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@levelchar}%
\ifx\null#3\null
\def\@gls@checklevel{\@gls@checklevel#2!|\null}%
\else
\def\@gls@checklevel{\@gls@checklevel#2!#3\null}%
\fi
\fi
\@gls@checklevel
}
\def\@gls@checkactual#1?#2?#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkactual{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar\@gls@quotechar\@gls@actualchar}%

```



```

\def\@gls@checkactual{\@gls@checkactual#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@gls@actualchar}%
\ifx\null#3\null
\def\@gls@checkactual{\@gls@checkactual#2??\null}%
\else
\def\@gls@checkactual{\@gls@checkactual#2?#3\null}%
\fi
\fi
\@gls@checkactual
}
\bggroup
\catcode'\0\relax
\catcode'\12\relax
\gdef\@gls@checkesc#1\#2\#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@checkesc{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@backslashchar\@gls@quotechar\@backslashchar}%
\def\@gls@checkesc{\@gls@checkesc#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\@gls@quotechar\@backslashchar}%
\ifx\null#3\null
\def\@gls@checkesc{\@gls@checkesc#2\\\null}%
\else
\def\@gls@checkesc{\@gls@checkesc#2\#3\null}%
\fi
\fi
\@gls@checkesc
}
\egroup
\def\@gls@xdycheckquote#1"#2"#3\null{%
\@gls@tmpb=\expandafter{\@gls@checkedmkidx}%
\toks@={#1}%
\ifx\null#2\null
\ifx\null#3\null
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@}%
\def\@gls@xdycheckquote{\relax}%
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"\string"}%

```

```

\def\@gls@xdycheckquote{\@gls@xdycheckquote#3\null}%
\fi
\else
\edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
\string"}%
\ifx\null#3\null
\def\@gls@xdycheckquote{\@gls@xdycheckquote#2""\null}%
\else
\def\@gls@xdycheckquote{\@gls@xdycheckquote#2"#3\null}%
\fi
\fi
\fi
\@gls@xdycheckquote
}
\edef\def@gls@xdycheckbackslash{%
\noexpand\def\noexpand\@gls@xdycheckbackslash##1\@backslashchar
##2\@backslashchar##3\noexpand\null{%
\noexpand\@gls@tmpb=\noexpand\expandafter
{\noexpand\@gls@checkedmkidx}%
\noexpand\toks@={##1}%
\noexpand\ifx\noexpand\null##2\noexpand\null
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{\relax}%
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar\@backslashchar\@backslashchar}%
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##3\noexpand\null}%
\noexpand\fi
\noexpand\else
\noexpand\edef\noexpand\@gls@checkedmkidx{%
\noexpand\the\noexpand\@gls@tmpb\noexpand\the\noexpand\toks@
\@backslashchar\@backslashchar}%
\noexpand\ifx\noexpand\null##3\noexpand\null
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
\@backslashchar\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@xdycheckbackslash{%
\noexpand\@gls@xdycheckbackslash##2\@backslashchar
##3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@xdycheckbackslash
}%
}
\def@gls@xdycheckbackslash

```

```

\newlength\gls@tmplen
\newcommand*{\glsdohypertarget}[2]{%
  \glsdoshowtarget{#1}%
  {%
    \glsmeasureheight{\gls@tmplen}{#2}%
    \raisebox{\gls@tmplen}{\hypertarget{#1}{}}%
    #2%
  }%
}
\newcommand*{\glsdohyperlink}[2]{%
  \glsdoshowtarget{#1}{\hyperlink{#1}{#2}}%
}
\newcommand*{\glsdonohyperlink}[2]{#2}
\ifcsundef{hyperlink}%
{%
  \let\@glslink\glsdonohyperlink
}%
{%
  \let\@glslink\glsdohyperlink
}
\ifcsundef{hypertarget}%
{%
  \let\@glstarget\@secondoftwo
}%
{%
  \let\@glstarget\glsdohypertarget
}
\newcommand{\glsdisablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glsdonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\glsenablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@gls@hyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}%
}
\def\@gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty

```

```

\def\glsinsert{#3}%
\def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glssunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
\new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}%
}
\def\@Gls@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glssunset{#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@glssunset{#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@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\glsifplural\@firstoftwo
  \let\glscapscase\@firstofthree
  \let\glscustomtext\@empty
  \def\glsinsert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \@gl@do@gl@unset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gl@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@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}%
    \@gl@do@gl@unset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\GLSpl}{\@gl@hyp@opt\@GLSpl}
\glsmfublocker{\GLSpl}
\newcommand*{\@GLSpl}[2] [] {%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2} []}%
}
\def\@GLSpl@#1#2[#3] {%
  \glsdoifexists{#2}%
  {%
    \let\do@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}%
    \@gl@do@gl@unset{#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\glsapscase\@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}}%
}
\glsfuaddmap{\glsdisp}{\Glsdisp}
\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{\%
  \glsdoifexists{#2}{\%
    {\%
      \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
      \@gls@link{#1}{#2}{#3}%
    }%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
\newcommand*{\@glstext}[2][\%
  \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}]
\def\@glstext@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\glsfublocker{\GLStext}
\newcommand*{\@GLStext}[2][\%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}]
\def\@GLStext@#1#2[#3]{\%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\glsfuaddmap{\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}{\glsupercase{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}
\newcommand*{\@glsplural}[2] [] {%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2} []}]
\def\@glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2] [] {%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2} []}]
\def\@Glsplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\Glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2] [] {%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2} []}]
\def\@GLSplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2} []}]
\def\@glsfirstplural@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2] [] {%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2} []}]
\def\@Glsfirstplural@#1#2[#3] {%

```

```

    \@gls@field@link{#1}{#2}{\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}{\glssupercase{\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}{\glssupercase{\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}{\glssupercase{\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}{\glsupercase{\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}
\glsmfublocker{\Glsymbol}
\newcommand*{\@Glsymbol}[2] [] {%
  \new@ifnextchar[{\@Glsymbol@{#1}{#2}}{\@Glsymbol@{#1}{#2} []}]
\def\@Glsymbol@#1#2[#3] {%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrysymbol{#2}#3}%
}
\newrobustcmd*{\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{\glssymbolplural}{\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}{\glssupercase{\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}{\glssupercase{\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}{\glssupercase{\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}{\glsupercase{\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}{\glsupercase{\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}{\glssupercase{\glssentryuserv{#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}{\glssentryuservi{#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}{\glssupercase{\glssentryuservi{#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{\glssentryshort{#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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsentryshort{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gl@hyp@opt\ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\ns@ACRshort}[2] []{%
\new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glssupercase{\acronymfont{\Glsentryshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrshortpl}{\@gl@hyp@opt\ns@acrshortpl}
\newcommand*{\ns@acrshortpl}[2] []{%
\new@ifnextchar[{\@acrshortpl{#1}{#2}}{\@acrshortpl{#1}{#2} []}%
}
\def\@acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%

```

```

\acronymfont{\glsentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrshortpl}{\@gls@hyp@opt\@ns@Acrshortpl}
\glsmfuaddmap{\Acrshortpl}{\Acrshortpl}
\newcommand*{\ns@Acrshortpl}[2][]{%
\new@ifnextchar[{\@Acrshortpl{#1}{#2}}{\@Acrshortpl{#1}{#2} []}%
}
\def\@Acrshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsentryshortpl{#2}}#3%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\@ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][]{%
\new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\acronymfont{\glsentryshortpl{#2}}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\acrlong}{\@gls@hyp@opt\@ns@acrlong}
\newcommand*{\ns@acrlong}[2][]{%
\new@ifnextchar[{\@acrlong{#1}{#2}}{\@acrlong{#1}{#2} []}%
}

```

```

\def\@acrlong#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glstrylong{#2}#3%
    }%
    \@gl@link{#1}{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@AcrLong{\@gl@hyp@opt\@ns@AcrLong}
\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\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsstrylong{#2}#3%
    }%
    \@gl@link{#1}{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@ACRLong{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glsupercase{\glstrylong{#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\glsupcase\@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\glsupcase\@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
  \def\glslabel{#2}%
  \let\glsifplural\@firstoftwo
  \let\glscapscase\@thirdofthree
  \let\glsinsert\@empty
  \def\glscustomtext{%
    \glssupercase{\glssentrylongpl{#2}#3}%
  }%
  \@gl@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gl@entry@field}[2]{%
  \csuse{glo@\glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsetentryfield}[3]{%
  \letcs{#1}{glo@\glsdetoklabel{#2}@#3}%
}
\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs{@glo@text}{glo@\glsdetoklabel{#1}@#2}%
    \ifdef{@glo@text}
    {%
      \expandafter\glssentencecase\expandafter{\@glo@text}%
    }%
    {%
      ??\PackageError{glossaries}{The field ‘#2’ doesn’t exist for glossary
        entry ‘\glsdetoklabel{#1}’}{Check you have correctly spelt the entry
        label and the field name}%
    }%
  }%
  {%
    ??%
  }%
}
\newcommand{\@Glsentryfield}[2]{%
  \glstexorpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@gl@entry@field{#1}{#2}}}%
}
\ifdef\texorpdfstring
{\newcommand{\glstexorpdfstring}{\texorpdfstring}}
{\newcommand{\glstexorpdfstring}[2]{#1}}
\newcommand*{\glsetentryname}[1]{\@gl@entry@field{#1}{name}}
\newcommand*{\Glsentryname}[1]{%
  \glstexorpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@gl@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
  }%
  {%
    \@Gls@entry@field{#1}{name}%
  }%
}
\newcommand*{\glsentrydesc}[1]{\@gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Glsentryfield{#1}{desc}}
\glsmfuaddmap{\glsentrydesc}{\Glsentrydesc}

```

```

\newcommand*{\glentrydescplural}[1]{%
  \@gls@entry@field{#1}{descplural}%
}
\newcommand*{\Glsentrydescplural}[1]{%
  \@Glsentryfield{#1}{descplural}%
}
\glsmfuaddmap{\glentrydescplural}{\Glsentrydescplural}
\newcommand*{\glentrytext}[1]{\@gls@entry@field{#1}{text}}
\newcommand*{\Glsentrytext}[1]{%
  \@Glsentryfield{#1}{text}%
}
\glsmfuaddmap{\glentrytext}{\Glsentrytext}
\newcommand*{\glentryplural}[1]{%
  \@gls@entry@field{#1}{plural}%
}
\newcommand*{\Glsentryplural}[1]{%
  \@Glsentryfield{#1}{plural}%
}
\glsmfuaddmap{\glentryplural}{\Glsentryplural}
\newcommand*{\glentrysymbol}[1]{%
  \@gls@entry@field{#1}{symbol}%
}
\newcommand*{\Glsentrysymbol}[1]{%
  \@Glsentryfield{#1}{symbol}%
}
\glsmfuaddmap{\glentrysymbol}{\Glsentrysymbol}
\newcommand*{\glentrysymbolplural}[1]{%
  \@gls@entry@field{#1}{symbolplural}%
}
\newcommand*{\Glsentrysymbolplural}[1]{%
  \@Glsentryfield{#1}{symbolplural}%
}
\glsmfuaddmap{\glentrysymbolplural}{\Glsentrysymbolplural}
\newcommand*{\glentryfirst}[1]{%
  \@gls@entry@field{#1}{first}%
}
\newcommand*{\Glsentryfirst}[1]{%
  \@Glsentryfield{#1}{first}%
}
\glsmfuaddmap{\glentryfirst}{\Glsentryfirst}
\newcommand*{\glentryfirstplural}[1]{%
  \@gls@entry@field{#1}{firstpl}%
}
\newcommand*{\Glsentryfirstplural}[1]{%
  \@Glsentryfield{#1}{firstpl}%
}
\glsmfuaddmap{\glentryfirstplural}{\Glsentryfirstplural}
\newcommand*{\glscapitalisewords}[1]{%
  \capitalisewords{#1}%
}

```

```

\newcommand*{\glstrytitlecase}[2]{%
  \glstexorpdfstring
    {\@glstrytitlecase{#1}{#2}}%
    {\MFUsentencecase{\@gls@entry@field{#1}{#2}}}%
}
\newrobustcmd*{\@glstrytitlecase}[2]{%
  \glstdoifexists{#1}%
  {%
    \glstfieldfetch{#1}{#2}{\@gls@value}%
    \expandafter\glscapitalisewords\expandafter{\@gls@value}%
  }%
}
\newcommand*{\glstrytype}[1]{\@gls@entry@field{#1}{type}}
\newcommand*{\glstrytype}[1]{%
  \@gls@entry@field{#1}{type}%
}
\newcommand*{\glstryparent}[1]{%
  \@gls@entry@field{#1}{parent}%
}
\newcommand*{\glstryuseri}[1]{%
  \@gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuseri}[1]{%
  \@Glsentryfield{#1}{useri}%
}
\glsmfuaddmap{\glstryuseri}{\Glsentryuseri}
\newcommand*{\glstryuserii}[1]{%
  \@gls@entry@field{#1}{userii}%
}
\newcommand*{\Glsentryuserii}[1]{%
  \@Glsentryfield{#1}{userii}%
}
\glsmfuaddmap{\glstryuserii}{\Glsentryuserii}
\newcommand*{\glstryuseriii}[1]{%
  \@gls@entry@field{#1}{useriii}%
}
\newcommand*{\Glsentryuseriii}[1]{%
  \@Glsentryfield{#1}{useriii}%
}
\glsmfuaddmap{\glstryuseriii}{\Glsentryuseriii}
\newcommand*{\glstryuseriv}[1]{%
  \@gls@entry@field{#1}{useriv}%
}
\newcommand*{\Glsentryuseriv}[1]{%
  \@Glsentryfield{#1}{useriv}%
}
\glsmfuaddmap{\glstryuseriv}{\Glsentryuseriv}
\newcommand*{\glstryuserv}[1]{%
  \@gls@entry@field{#1}{userv}%
}

```

```

\newcommand*{\Glsentryuser}[1]{%
  \@Glsentryfield{#1}{user}%
}
\glsmfuaddmap{\Glsentryuser}{\Glsentryuser}
\newcommand*{\Glsentryuseri}[1]{%
  \@Gls@entry@field{#1}{useri}%
}
\newcommand*{\Glsentryuservi}[1]{%
  \@Glsentryfield{#1}{uservi}%
}
\glsmfuaddmap{\Glsentryuservi}{\Glsentryuservi}
\newcommand*{\Glsentryshort}[1]{\@Gls@entry@field{#1}{short}}
\newcommand*{\Glsentryshorti}[1]{%
  \@Glsentryfield{#1}{short}%
}
\glsmfuaddmap{\Glsentryshort}{\Glsentryshort}
\newcommand*{\Glsentryshortpl}[1]{\@Gls@entry@field{#1}{shortpl}}
\newcommand*{\Glsentryshortpli}[1]{%
  \@Glsentryfield{#1}{shortpl}%
}
\glsmfuaddmap{\Glsentryshortpl}{\Glsentryshortpl}
\newcommand*{\Glsentrylong}[1]{\@Gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylongi}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuaddmap{\Glsentrylong}{\Glsentrylong}
\newcommand*{\Glsentrylongpl}[1]{\@Gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpli}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuaddmap{\Glsentrylongpl}{\Glsentrylongpl}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\Glsentryshort{#1})%
}
\newcommand*{\Glsentryfulli}[1]{%
  \Glsentrylong{#1} (\Glsentryshort{#1})%
}
\glsmfuaddmap{\Glsentryfull}{\Glsentryfull}
\newcommand*{\Glsentryfullpl}[1]{%
  \acrfullformat{\Glsentrylongpl{#1}}{\acronymfont{\Glsentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpli}[1]{%
  \Glsentrylongpl{#1} (\Glsentryshortpl{#1})%
}
\glsmfuaddmap{\Glsentryfullpl}{\Glsentryfullpl}
\newcommand*{\Glsentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@Gls@entry@field{#1}{numberlist}%
  }%
}

```

```

}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glsentrynumberlist\space instead%
    }%
    \glsentrynumberlist{#1}%
  }%
}%
{%
\newcommand*{\glsdisplaynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \bgroup
      \protected@edef\@glo@label{\glsdetoklabel{#1}}%
      \let\@org@glsnumberformat\glsnumberformat
      \def\glsnumberformat##1{##1}%
      \protected@edef\the@numberlist{%
        \csname glo@\@glo@label @numberlist\endcsname}%
      \def\@gls@numlist@sep{%
      \def\@gls@numlist@nextsep{%
      \def\@gls@numlist@lastsep{%
      \def\@gls@thislist{%
      \def\@gls@donext@def{%
      \renewcommand\do[1]{%
        \protected@edef\@gls@thislist{%
          \@gls@thislist
          \noexpand\@gls@numlist@sep
          ##1%
        }%
        \let\@gls@numlist@sep\@gls@numlist@nextsep
        \def\@gls@numlist@nextsep{\glsnumlistsep}%
        \@gls@donext@def
        \def\@gls@donext@def{%
          \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
        }%
      }%
      \expandafter \glsnumlistparser \expandafter{\the@numberlist}%
      \let\@gls@numlist@sep\@gls@numlist@lastsep
      \@gls@thislist
    \egroup
  }%
}%
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ & }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%

```

```

\def\@glo@label{#2}%
\@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*\glsadd[2][]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo@glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\glsmfuexcl{\glsadd}
\newcommand*\@gls@adjustmode{}
\AtBeginDocument{\renewcommand*\@gls@adjustmode{\ifvmode\mbox{}\fi}}
\newrobustcmd*\glsaddall[1][]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*\glsaddallunused[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*\glsignore[1]{}
\edef\glsopenbrace{\expandafter\@gobble\string\{ }
\edef\glsclosebrace{\expandafter\@gobble\string\} }
\edef\glsbackslash{\expandafter\@gobble\string\ }
\edef\glsquote#1{\string"#1\string"}
\edef\glspercentchar{\expandafter\@gobble\string\% }
\edef\glsstildechar{\string~ }
\ifglsxindy
  \newcommand*\@glsfirstletter{A}
\fi
\newcommand*\GlsSetXdyFirstLetterAfterDigits{%
  \ifstar\s@GlsSetXdyFirstLetterAfterDigits@GlsSetXdyFirstLetterAfterDigits}
\ifglsxindy
  \newcommand*\@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter{#1}}
  \newcommand*\s@GlsSetXdyFirstLetterAfterDigits[1]{%
    \renewcommand*\@glsfirstletter{#1}}

```

```

        \@onelevel@sanitize\@glsfirstletter
    }
\else
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
        \glsnoxywarning\GlsSetXdyFirstLetterAfterDigits}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}{%
        \@GlsSetXdyFirstLetterAfterDigits
    }
\fi
\ifglxsindy
    \newcommand*{\@xdynumbergrouporder}{:before \string"\@glsfirstletter\string"}
\fi
\newcommand*{\GlsSetXdyNumberGroupOrder}[1]{%
    \@ifstar\s@GlsSetXdyNumberGroupOrder\@GlsSetXdyNumberGroupOrder
}
\ifglxsindy
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \renewcommand*{\@xdynumbergrouporder}{#1}%
    }
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}[1]{%
        \renewcommand*{\@xdynumbergrouporder}{#1}%
        \@onelevel@sanitize\@xdynumbergrouporder
    }
\else
    \newcommand*{\@GlsSetXdyNumberGroupOrder}[1]{%
        \glsnoxywarning\GlsSetXdyNumberGroupOrder}
    \newcommand*{\s@GlsSetXdyNumberGroupOrder}{%
        \@GlsSetXdyNumberGroupOrder}
\fi
\newcommand*{\@glsminrange}{2}
\ifglxsindy
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \renewcommand*{\@glsminrange}{#1}}
\else
    \newcommand*{\GlsSetXdyMinRangeLength}[1]{%
        \glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglxsindy
    \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
  \glsoopenbrace\string"^^J\space\space\space
  :close-head \string"\glsclosebrace\string")}%
\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}%
\write\glswrite{@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\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{\glspcrcentchar\space makeindex style file
      created by the glossaries package}
    \write\glswrite{\glspcrcentchar\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"glsnnumbers\string"}
\write\glswrite{page_compositor \string"glscompositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
    \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
    \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}

```

```

\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{%
\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####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\noexpand\null}%
\noexpand\else
\noexpand\def\noexpand\@gls@checkquote{%
\noexpand\@gls@checkquote####2####3\noexpand\null}%
\noexpand\fi
\noexpand\fi
\noexpand\@gls@checkquote
}%
}%

```

```

\@gls@docheckquotedef
\edef\@gls@docheckquotedef{%
  \noexpand\renewcommand{\noexpand\@gls@checkmkidxchars}[1]{%
    \noexpand\def\noexpand\@gls@checkedmkidx{%
      \noexpand\expandafter\noexpand\@gls@checkquote####1\noexpand\@nil
      #1#1\noexpand\null
      \noexpand\expandafter\noexpand\@gls@updatechecked
      \noexpand\@gls@checkedmkidx{####1}%
      \noexpand\def\noexpand\@gls@checkedmkidx{%
        \noexpand\expandafter\noexpand\@gls@checkescquote####1\noexpand\@nil
        \expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
        \noexpand\null
        \noexpand\expandafter\noexpand\@gls@updatechecked
        \noexpand\@gls@checkedmkidx{####1}%
        \noexpand\def\noexpand\@gls@checkedmkidx{%
          \noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
          \noexpand\?\noexpand\?\noexpand\null
          \noexpand\expandafter\noexpand\@gls@updatechecked
          \noexpand\@gls@checkedmkidx{####1}%
          \noexpand\def\noexpand\@gls@checkedmkidx{%
            \noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
            \noexpand?\noexpand?\noexpand\null
            \noexpand\expandafter\noexpand\@gls@updatechecked
            \noexpand\@gls@checkedmkidx{####1}%
            \noexpand\def\noexpand\@gls@checkedmkidx{%
              \noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
              \noexpand|\noexpand|\noexpand\null
              \noexpand\expandafter\noexpand\@gls@updatechecked
              \noexpand\@gls@checkedmkidx{####1}%
              \noexpand\def\noexpand\@gls@checkedmkidx{%
                \noexpand\expandafter\noexpand\@gls@checkescbar####1\noexpand\@nil
                \noexpand\|\noexpand\|\noexpand\null
                \noexpand\expandafter\noexpand\@gls@updatechecked
                \noexpand\@gls@checkedmkidx{####1}%
                \noexpand\def\noexpand\@gls@checkedmkidx{%
                  \noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
                  \noexpand!\noexpand!\noexpand\null
                  \noexpand\expandafter\noexpand\@gls@updatechecked
                  \noexpand\@gls@checkedmkidx{####1}%
                  \noexpand\def\noexpand\@gls@checkedmkidx{%
                    \noexpand\expandafter\noexpand\@gls@checksclevel####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 \listfilename\space
            -t \jobname.\csuse{@glotype@#1@log}
            -o \jobname.\csuse{@glotype@#1@in}
            \jobname.\csuse{@glotype@#1@out}}}%
        }%
        \@gls@dothiswrite
    }%
    {%
        \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
    }%
}
\fi
\ifglxsindy
\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{@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}}%
}%
}
\else
\newcommand*{\@gls@automake@immediate}[1]{%
\ifglossaryexists{#1}
{%
\IfFileExists{\jobname.\csuse{@gls@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{@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
\ifglssavenumberlist
\edef\@gls@dodolistparser{\noexpand\DeclareListParser

```

```

        {\noexpand\glsnumlistparser}{\delimN}}}%
    \@gls@dodeflistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
    \renewcommand*{\@gls@doautomake}{%
        \protected@edef\gls@tmp{\@gls@automake@types}%
        \@for\@gls@type:=\gls@tmp\do{%
            \ifdefempty{\@gls@type}{}%
            {\@gls@automake{\@gls@type}}}%
        }%
    }%
\fi
\@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}
\newcommand\@makeglossaries@warn@noprintglossary{%
    \ifdefstring{\@glo@types}{,}%
    {%
        \GlossariesWarningNoLine{No glossaries have been defined}%
    }%
    {%
        \GlossariesWarningNoLine{No \string\printglossary\space
            or \string\printglossaries\space
            found. ^^J(Remove \string\makeglossaries\space if you
            don't want any glossaries.) ^^JThis document will not
            have a glossary}%
    }%
}%
\AtEndDocument{%
    \warn@nomakeglossaries
    \warn@noprintglossary
}
\newcommand*{\makenoidxglossaries}{%
    \def\glsindexingsetting{noidx}%
    \@domakeglossaries
    {%
        \renewcommand{\@gls@noref@warn}[1]{%
            \GlossariesWarning{Empty glossary for
                \string\printnoidxglossary[type={##1}].
                Rerun may be required (or you may have forgotten to use
                commands like \string\gls)}%
        }%
        \let\@gls@checkmkidxchars\@gobble
        \glsesclocationsfalse
        \let\@do@@wrglossary\gls@noidxglossary
    }
}

```

```

\let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
\let\gls@checkseeallowed\relax
\renewcommand{\@do@seeglossary}[2]{%
  \protected@edef\@gls@label{\glsdetoklabel{##1}}%
  \protected@write\@auxout{}\{%
    \string\@gls@reference
      {\csname glo@\@gls@label @type\endcsname}%
      {\@gls@label}%
      {%
        \string\glsseeformat##2}%
      }%
  }%
}%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J
    found. (Remove \string\makenoidxglossaries\space if you
    don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for '##1'. Either
        a rerun is required or you haven't referenced the entry}%
    }%
  }%
}%

```

```

\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@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@glssееformat\glssееformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glssееformat##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\glssееformat\@gls@org@glssееformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries

```

```

\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*\@glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*\@glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*\@no@makeglossaries{%
  \PackageError{glossaries}{You can't use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*\@gls@noidxglossary{%
  \protected@write\@auxout{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
     {\@glo@counterprefix}%
     {\@gls@counter}%
     {\@glsnumberformat}%
     {\@glslocref}}%
    }%
  }%
}
}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*\@glswritefiles{%
  \foralllglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '\@glo@type' has no

```

```

        entries.^^JRemember to use package option ‘nomain’ if
you
        don’t want to^^Juse the main glossary}%
    \else
        \GlossariesWarningNoLine{Glossary ‘\@glo@type’ has no
            entries}%
    \fi
\else
    \@glsopenfile{\glswrite}{\@glo@type}%
    \immediate\write\glswrite{%
        \expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    \immediate\closeout\glswrite
\fi
}%
}
\newcommand*{\gls@glossary}[1]{%
    \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
    \if@gls@debug
        \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
    \fi
}
\newcommand{\@gls@renewglossary}{%
    \gdef\@gls@glossary##1{\@bsphack\begin@group\gls@wrglossary{##1}}%
    \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
    \ifglssavewrites
        \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
        \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
            \expandafter{\@gls@tmp^^J}%
    \else
        \ifcsdef{glo@#1@file}%
        {%
            \expandafter\protected@write\csname glo@#1@file\endcsname{%
                \gls@disablepagerefexpansion}{#2}%
            }%
        {%
            \ifignoredglossary{#1}{}%
            {%
                \GlossariesWarning{No file defined for glossary ‘#1’}%
            }%
        }%
    \fi
\endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
    \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%

```



```

}
\newcommand*{\glswriteentry}[2]{%
  \ifglindexonlyfirst
    \ifglused{#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
\ifglsclocations
  \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]{%
  \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@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
}
\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]{%
  \ifglswrallowprimitivemods
    \@@do@esc@wrglossary{#1}%
  \else
    \@@do@noesc@wrglossary{#1}%
  \fi
}
\newcommand{\glswrglossdisableanchorcmds}{\let\glstexorpdfstring\@secondoftwo}
\ifdef\texorpdfstring
  {\appto\glswrglossdisableanchorcmds{\let\texorpdfstring\@secondoftwo}}{}
\ifdef\pdfstringdefPreHook
  {\appto\glswrglossdisableanchorcmds{\pdfstringdefPreHook}}{}
\newcommand{\glswrglosslocationtarget}[1]{#1}
\newcommand{\glswrglosslocationtextfmt}[1]{#1}
\newcommand*{\@@do@noesc@wrglossary}[1]{%
  \expandafter\def\expandafter\@glslocref\expandafter{\theglsentrycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglentrycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glstetoklabel{#1}}%
  \@@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
      \def\number{\expandafter\gls@number}%
      \def\romannumeral##1{%
        \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
    \fi
    \def\@arabic##1{%
      \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
    \def\@Roman##1{%

```

```

\ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
\def\@alph##1{%
\ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
\def\@Alph##1{%
\ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
\@wrglossarynumberhook
\gls@disablepagerefexpansion
\protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\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}{\@glslocref}}%
\fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%

```

```

\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\@do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{%
}else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{%
}else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been^^J
referenced but does not exist”}%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglxsindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:key (\csname glo@#1@index\endcsname)
:xref (\string"\@gls@xref\string")
:attr \string"see\string"
)
}%
}else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string@glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
}else

```

```

\def#1{{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*{\glssee}[3][\seename]{%
\do@seeglossary{#2}{[#1]{#3}}
\newcommand*{\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*{\glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}
\newrobustcmd*{\glsseelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext
\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseelastsep
\let\@gls@donext\glsseesep
}%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }
\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{%
\ifglshasshort{#1}{\glentrytext{#1}}{\glentryname{#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=glndefaulttype]{%
  \let\@printgloss@checkexists\@printgloss@checkexists@noignored
  \@printglossary{#1}{\@print@glossary}%
}
\newcommand*{\printglossaries}{%
  \forallglossaries{\@glo@type}{\printglossary[type=\@glo@type]}%
}
\newcommand*{\printnoidxglossary}[1][type=glndefaulttype]{%
  \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{glndefaulttype}%
  \def\glossarytitle{\csname @glo@type\@glo@type @title\endcsname}%
  \def\glossarytoctitle{\glossarytitle}%
  \let\org@glossarytitle\glossarytitle
  \def\@glossarystyle{%
    \ifx\@glossary@default@style\relax
      \GlossariesWarning{No default glossary style provided \MessageBreak
        for the glossary ‘\@glo@type’. \MessageBreak
        Using fallback. \MessageBreak
        To fix this set the style with \MessageBreak
        \string\setglossarystyle\space or use the \MessageBreak
        style key=value option}%
    \fi
  }%
  \def\gls@dotoc@title{\gls@set@toc@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 @glotype@\@glo@type @title\endcsname
\glossarytitle
\fi
\let\currentglossary\@glo@type
\let\org@glossaryentrynumbers\glossaryentrynumbers
\let\glsnonextpages\@glsnonextpages
\let\glsnextpages\@glsnextpages
\let\nopostdesc\@nopostdesc
\gls@dotoc@title
\@glossarystyle
\let\gls@org@glossaryentryfield\glossentry
\let\gls@org@glossarysubentryfield\subglossentry
\renewcommand{\glossentry}[1]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##1}}%
\gls@org@glossaryentryfield{##1}%
}%
\renewcommand{\subglossentry}[2]{%
\protected@xdef\glscurrententrylabel{\glsdetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\egroup
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprintglossary\relax
}
\newcommand{\@print@glossary}{%
\makeatletter
\@input{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
\IfFileExists{\jobname.\csname @glotype@\@glo@type @in\endcsname}%
{}%
{\null}%
\ifglxindy
\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@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}{@glssref@\@glo@type}%
  {%
  }%
  {%
    \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
  }%
}%
{%
  \@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
  \dtlinsertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
  \letcs\@glssort@A{glo@glstdetoklabel{#1}@sort}%
  \letcs\@glssort@B{glo@glstdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@glssort@B}%
    {\expandonce\@glssort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
  \letcs\@glssort@A{glo@glstdetoklabel{#1}@sort}%
  \letcs\@glssort@B{glo@glstdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%
    \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
    {\expandonce\@glssort@B}%
    {\expandonce\@glssort@A}%
  }%
  \glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
  \letcs\@glssort@A{glo@glstdetoklabel{#1}@sort}%
  \letcs\@glssort@B{glo@glstdetoklabel{#2}@sort}%
  \edef\glo@do@compare{%

```

```

        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{glo@\glsdetoklabel{#2}@sort}%
    \edef\glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \glo@do@compare
}
\newcommand*{\@glo@sortmacro@word}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@word}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=word]}{}%
    }%
}
\newcommand*{\@glo@sortmacro@letter}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
            \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
            \string\printnoidxglossary[sort=letter]}{}%
    }%
}
\newcommand*{\@glo@sortmacro@standard}[1]{%
    \ifdefstring{\@glo@default@sorttype}{standard}%
    {%
        \ifcsdef{\@glo@sorthandler@\glsorder}%
        {%
            \@glo@sortentries{\csuse{\@glo@sorthandler@\glsorder}}{#1}%
        }%
        {%
            \PackageError{glossaries}{Unknown sort handler '\glsorder'}{}%
        }%
    }%
    {%

```

```

\PackageError{glossaries}{Conflicting sort options:^^J
\string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
\string\printnoidxglossary[sort=standard]}{}%
}%
}
\newcommand*{\@glo@sortmacro@case}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
\@glo@sortentries{\@glo@sorthandler@case}{#1}%
}%
{%
\PackageError{glossaries}{Conflicting sort options:^^J
\string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
\string\printnoidxglossary[sort=case]}{}%
}%
}
\newcommand*{\@glo@sortmacro@nocase}[1]{%
\ifdefstring{\@glo@default@sorttype}{standard}%
{%
\@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
}%
{%
\PackageError{glossaries}{Conflicting sort options:^^J
\string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
\string\printnoidxglossary[sort=nocase]}{}%
}%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
\def\@glo@sortinglist{}%
\forlslentries[#1]{\@gls@thislabel}%
{%
\xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
{%
\listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
}%
}%
}%
\cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
\ifinlistcs[#1]{\@glsref@\@glo@type}%
{}%
{%
\listcsadd{\@glsref@\@glo@type}{#1}%
}%
\ifcsdef{\@glo@sortingchildren@#1}%
{%
\@glo@addchildren{\@glo@type}{#1}%
}%
}

```

```

    {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@\@glo@type}%
  {%
    \ifcsdef{\@glo@sortmacro@\@glo@sorttype}%
    {%
      \csuse{\@glo@sortmacro@\@glo@sorttype}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler '\@glo@sorttype'}{ }%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
    \begin{theglossary}%
    \glossaryheader
    \glsresetentrylist
    \forlistcsloop{\@gls@noidx@do}{\@glsref@\@glo@type}%
    \end{theglossary}%
    \glossarypostamble
  }%
  {%
    \@gls@noref@warn{\@glo@type}%
  }%
}
}
\def\@glo@grabfirst#1#2\@nil{%
  \def\@gls@firsttok{#1}%
  \ifdefempty\@gls@firsttok
  {%
    \def\@glo@thislettergrp{0}%
  }%
  {%
    \@onelevel@sanitize\@gls@firsttok
    \expandafter\@glo@grabfirst\@gls@firsttok{ }{\@nil
  }%
}
\def\@glo@grabfirst#1#2\@nil{%
  \ifdefempty\@glo@thislettergrp
  {%
    \def\@glo@thislettergrp{glssymbols}%
  }%
  {%
    \count@=\uccode' #1\relax
    \ifnum\count@=0\relax
      \def\@glo@thislettergrp{glssymbols}%
    \else
      \ifdefstring\@glo@sorttype{case}%
      {%

```

```

        \count@=#1\relax
    }%
    {%
    }%
    \edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{glo@glsdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{glo@glsdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%
}%
{%
\subglossentry{\gls@level}{#1}%
{%
\glsnoidxprenumberlist{#1}%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%
{%
\letcs{\@gls@sort}{glo@glsdetoklabel{#1}@sort}%
\expandafter\glo@grabfirst\@gls@sort{}\@nil
\ifdefequal{\@glo@thislettergrp}{\@gls@currentlettergroup}%
{}%
{%
\ifdefempty{\@gls@currentlettergroup}{}%
{%
\global\let\@glo@thislettergrp\@glo@thislettergrp
\glsgroupskip
}%
\glsgroupheading{\@glo@thislettergrp}%
}%
\global\let\@gls@currentlettergroup\@glo@thislettergrp
\ifdefvoid{\@gls@loclist}
{%
\glossentry{#1}{}%
}%
{%
\glossentry{#1}%
{%
\glsnoidxprenumberlist{#1}%
\glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
}%
}%
}%

```

```

    }%
}
\newcommand*{\glsnoidxloclist}[1]{%
  \def\@gls@noidxloclist@sep{}%
  \def\@gls@noidxloclist@prev{}%
  \forlistloop{\glsnoidxloclisthandler}{#1}%
}
\newcommand*{\glsnoidxloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    #1%
    \def\@gls@noidxloclist@sep{\delimN}%
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*{\glsnoidxdisplayloclisthandler}[1]{%
  \ifdefstring{\@gls@noidxloclist@prev}{#1}%
  {%
  }%
  {%
    \@gls@noidxloclist@sep
    \@gls@noidxloclist@prev
    \def\@gls@noidxloclist@prev{#1}%
  }%
}
\newcommand*\glsnoidxdisplayloc[4]{%
  \setentrycounter[#1]{#2}%
  \csuse{#3}{#4}%
}
\newcommand*{\@gls@reference}[3]{%
  \glsdoifexistsorwarn{#2}%
  {%
    \ifcsundef{\@glsref@#1}{\csgdef{\@glsref@#1}{}}{}%
    \ifinlistcs{#2}{\@glsref@#1}%
    {}%
    {\listcsgadd{\@glsref@#1}{#2}}%
    \ifcsundef{glo@glstdetoklabel{#2}@loclist}%
    {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
    {}%
    \listcsgadd{glo@glstdetoklabel{#2}@loclist}{#3}%
  }%
}
\define@key{printgloss}{type}{\def\@glo@type{#1}}
\define@key{printgloss}{title}{%
  \def\glossarytitle{#1}%
  \let\gls@dotocitle\relax
}

```

```

\define@key{printgloss}{toctitle}{%
\def\glossarytoctitle{#1}%
\let\gls@dottoctitle\relax
}
\define@key{printgloss}{style}{%
\ifcsundef{@glsstyle@#1}%
{%
\PackageError{glossaries}%
{Glossary style ‘#1’ undefined}{}%
}%
{%
\def\@glossarystyle{\setglossarystyle{#1}}%
}%
}
\define@choicekey{printgloss}{numberedsection}%
[\gls@numberedsection@val\gls@numberedsection@nr]%
{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{}%
\or
\renewcommand*{\@glossarysecstar}{}%
\renewcommand*{\@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@glossarysecstar}{*}%
\renewcommand*{\@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glssubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%

```

```

\ifglsonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi

```



```

}
\newcommand*{\glsrefentry}[1]{%
  \ifglentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else
    \ifglssubentrycounter
      \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
    \else
      \gls{#1}%
    \fi
  \fi
}
\newcommand*{\glentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}
\newcommand*{\glentryitem}[1]{%
  \ifglentrycounter
    \glsstepentry{#1}\glentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glsstepsubentry{#1}\glssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\newcommand*{\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }
}

```

```

    }%
}
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsnamefont{\Glsentryname{#1}}%
  }%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*{\glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrydesc{#1}%
  }%
}
\newcommand*{\Glossentrydesc}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrydesc{#1}%
  }%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*{\glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \glsentrysymbol{#1}%
  }%
}
\newcommand*{\Glossentrysymbol}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \Glsentrysymbol{#1}%
  }%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*{\glsgroupskip}{\}
\newcommand*{\glsgroupheading}[1]{\}
\newrobustcmd*{\glsgetgrouptitle}[1]{%
  \@gls@getgrouptitle{#1}{\@gls@grptitle}%
  \@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
  \def#2{#1}%
  \regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
  { \ifcsundef{#1groupname}{\letcs#2{#1groupname}} }
}
\ExplSyntaxOff
\newcommand*{\@gls@noidx@getgrouptitle}[2]{%

```

```

\DTLifint{#1}%
{\edef#2{\char#1\relax}}%
{%
  \ifcsundef{#1groupname}{\def#2{#1}}{\letcs#2{#1groupname}}%
}%
}
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\newcommand*{\setentrycounter}[2][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{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%

```

```

\PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
}%
{%
\csdef{@glsstyle@#1}{#2}%
}%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
\def\glshypernumber#1{#1}%
}%
{%
\def\glshypernumber#1{\@xp@glshypernumber{#1}}
}
\newrobustcmd{\@xp@glshypernumber}[1]{%
\@glshypernumber#1\nohyperpage{ }\@nil
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
\ifx\#1\%
\else
\gls@delimR#1\delimR\delimR\%
\fi
\ifx\#2\%
\else
#2%
\fi
\ifx\#3\%
\else
\@glshypernumber#3\@nil
\fi
}
\def\gls@delimR#1\delimR #2\delimR #3\%
\ifx\#2\%
\gls@delimN{#1}%
\else
\@gls@numberlink{#1}\delimR\@gls@numberlink{#2}%
\fi
}
\def\gls@delimN#1{\gls@delimN#1\delimN \delimN\}
\def\gls@delimN#1\delimN #2\delimN#3\%
\ifx\#3\%
\@gls@numberlink{#1}%
\else
\@gls@numberlink{#1}\delimN\@gls@numberlink{#2}%
\fi
}
\def\@gls@numberlink#1{%
\begingroup
\toks@={}%
\@gls@removespaces#1 \@nil

```

```

\endgroup
}
\def\@gls@removespaces#1 #2\@nil{%
\toks@=\expandafter{\the\toks@#1}%
\ifx\#2\%
\edef\@gls@x{\the\toks@}%
\ifx\@gls@x\empty
\else
\bgroup
\glswrglossdisableanchorcmds
\protected@edef\@gls@tmp{\glshypentrycounter\@gls@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{\noexpand\glswrglosslocationtextfmt{\the\toks@}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glshypernumber{#1}}}
\newcommand\oldacronym[4][\@gls@label]{%
\def\@gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}%
}%
}%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}\noexpand\xspace}{%
\noexpand\gls{#1}\noexpand\xspace}%
}%
}

```

```

}%
}
\newcommand{\newacronym}[4][\%
\gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
\renewcommand{\gls@newacronym}[4]{%
\PackageError{glossaries}{incompatible acronym style:
rollback may be required}%
{something has gone wrong with \string\newacronym. Try
\string\usepackage{glossaries}[=v4.49]}%
}%
\setacronymstyle{long-short}%
\newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
\acrfullformat{#1}{#3}{#4}[#5]{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
\newrobustcmd*{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
\acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
\acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
\acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
\acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
\acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
\acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
\newcommand*{\ns@acrfull}[2][\%
\new@ifnextchar[\@acrfull{#1}{#2}}%
{\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%

```

```

\acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}[]}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}[]}%
}
\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfullpl}{\@gls@hyp@opt\ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfullpl}{\@gls@hyp@opt\ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}%
    {\@Acrfullpl{#1}{#2}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfullpl}{\@gls@hyp@opt\ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}%
    {\@ACRfullpl{#1}{#2}[]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*{\acrnameformat}[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok

```

```

\newtoks\glslongtok
\newcommand*{\newacronymhook}{}
\newcommand*{\SetGenericNewAcronym}{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][{}]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
    \newacronymhook
  }%
  \protected@edef\@do@newglossaryentry{%
    \noexpand\newglossaryentry{\the\glslabeltok}%
    {%
      type=\acronymtype,%
      name={\expandonce{\acronymentry{##2}}},%
      sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
      text={\the\glsshorttok},%
      short={\the\glsshorttok},%
      shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
      long={\the\glslongtok},%
      longplural={\the\glslongtok\noexpand\glspluralsuffix},%
      \GenericAcronymFields,%
      \the\glskeylisttok
    }%
  }%
  \@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glssupercase{\genacrfullformat{##2}{##3}}}%
}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glssupercase{\genplacrfullformat{##2}{##3}}}%
}%
\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@}{(\firstacronymfont{\glsentryshort{##1}})}%
  \ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
  \ifglshaslong{\glslabel}{\glsacspace}{\glsacspace}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\genacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\Genacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshort{##1}}##2\space
    (\glsentrylong{##1})%
  }%
  \renewcommand*{\genplacrfullformat}[2]{%
    \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\Genplacrfullformat}[2]{%
    \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
    (\glsentrylongpl{##1})%
  }%
  \renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
  \renewcommand*{\acronymsort}[2]{##1}%
  \renewcommand*{\acronymfont}[1]{##1}%
  \renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
}

```

```

\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%

```

```

\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \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
      {%
        \glentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \Glentrylongpl{\glslabel}\glsinsert
      }%
      {%
        \glsuppercase
        {\glentrylongpl{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glscapscase
    {%
      \glentrylong{\glslabel}\glsinsert
    }%
    {%
      \Glentrylong{\glslabel}\glsinsert
    }%
    {%
      \glsuppercase
      {\glentrylong{\glslabel}\glsinsert}%
    }%
  }%
}
{%
  \glsgenentryfmt
}
{%
  {\glscustomtext\glsinsert}%
}
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glentrylong{##2}##3\space
      (\acronymfont{\glentryshort{##2}})}%
  }
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glentrylong{##2}##3\space
      (\acronymfont{\glentryshort{##2}})}%
  }
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glsuppercase{\glentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##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}{%
\glssupercase{\glsentrylongpl{##2}##3\space
(\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
\glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
\Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}

```

```

}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glentryshortpl{##1}}##2%
  \protect\footnote{\glentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
  \protect\footnote{\glentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glentryshort{##2}}##3\space
    (\glentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
    (\glentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsupercase{\acronymfont{\glentryshort{##2}}##3\space
      (\glentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\glentryshortpl{##2}}##3\space
    (\glentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
    (\glentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsupercase{\acronymfont{\glentryshortpl{##2}}##3\space
      (\glentrylongpl{##2})}}}%
\renewcommand*{\glentryfull}[1]{%
  \acronymfont{\glentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
  \acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{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]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
\let\acs\acrshort
\let\Acs\Acrshort
\let\acsp\acrshortpl
\let\Acsp\Acrshortpl

```

```

\let\acl\acrlong
\let\aclp\acrlongpl
\let\Acl\Acrlong
\let\Aclp\Acrlongpl
\let\acf\acrfull
\let\acfp\acrfullpl
\let\Acf\Acrfull
\let\Acfp\Acrfullpl
\let\ac\gls
\let\Ac\Gls
\let\acp\glspl
\let\Acp\Glspl
}
\ifglsacrshortcuts
\DefineAcronymSynonyms
\fi
\newcommand*{\SetDefaultAcronymDisplayStyle}[1]{%
\defglsentryfmt[#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@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouseri}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useri\endcsname
}
\newcommand*{\showglouserii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showglongame}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@name\endcsname
}
\newcommand*{\showglodesc}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@desc\endcsname
}

```

```

\newcommand*{\showglodescplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@descplural\endcsname
}
\newcommand*{\showglosort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@sort\endcsname
}
\newcommand*{\showglosymbol}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbol\endcsname
}
\newcommand*{\showglosymbolplural}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@symbolplural\endcsname
}
\newcommand*{\showgloshort}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@short\endcsname
}
\newcommand*{\showglolong}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@long\endcsname
}
\newcommand*{\showgloindex}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@index\endcsname
}
\newcommand*{\showgloflag}[1]{%
  \expandafter\show\csname ifglo@glstdetoklabel{#1}@flag\endcsname
}
\newcommand*{\showgloclist}[1]{%
  \expandafter\show\csname glo@glstdetoklabel{#1}@loclist\endcsname
}
\newcommand*{\showglofield}[2]{%
  \csshow{glo@glstdetoklabel{#1}@#2}%
}
\newcommand*{\showacronymlists}{%
  \show\@glsacronymlists
}
\newcommand*{\showglossaries}{%
  \show\@glo@types
}
\newcommand*{\showglossaryin}[1]{%
  \expandafter\show\csname @glotype@#1@in\endcsname
}
\newcommand*{\showglossaryout}[1]{%
  \expandafter\show\csname @glotype@#1@out\endcsname
}
\newcommand*{\showglossarytitle}[1]{%
  \expandafter\show\csname @glotype@#1@title\endcsname
}
\newcommand*{\showglossarycounter}[1]{%
  \expandafter\show\csname @glotype@#1@counter\endcsname
}
\newcommand*{\showglossaryentries}[1]{%
  \expandafter\show\csname glolist@#1\endcsname
}

```

}

8.36 Rollback v4.52 (glossary-hypernav-2022-11-03.sty)

Version 4.52 preserved for rollback.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{glossary-hypernav}[2022/11/03 v4.52 (NLCT)]
\newcommand*{\glsnavhyperlink}[3][\currentglossary]{%
  \protected@edef\gls@grplabel{#2}\protected@edef\gls@grptitle{#3}%
  \glslink{\glsnavhyperlinkname{#1}{#2}}{#3}%
}
\newcommand*{\glsnavhyperlinkname}[2]{\glsn:#1@#2}
\newcommand*{\glsnavhypertarget}{\protect\@gls@navhypertarget}
\newcommand*{\@gls@navhypertarget}[3][\currentglossary]{%
  \glsnavhypertarget{#1}{#2}{#3}%
}
\newcommand*{\@gls@navhypertarget}[3]{%
  \protected@write\@auxout{}\string\@gls@hypergroup{#1}{#2}}%
  \gls@target{\glsnavhyperlinkname{#1}{#2}}{#3}%
  \expandafter\let
    \expandafter\@gls@list\csname @gls@hypergroup@list@#1\endcsname
  \@for\@gls@elem:=\@gls@list\do{%
    \ifthenelse{\equal{\@gls@elem}{#2}}{\@endfortrue}{}%
  }
  \if@endfor
  \else
    \GlossariesWarningNoLine{Navigation panel
      for glossary type ‘#1’^^Jmissing group ‘#2’}%
    \gdef\gls@hypergroup@rerun{%
      \GlossariesWarningNoLine{Navigation panel
        has changed. Rerun LaTeX}}%
  \fi
}
\let\gls@hypergroup@rerun\relax
\AtEndDocument{\gls@hypergroup@rerun}
\newcommand*{\@gls@hypergroup}[2]{%
  \ifcsdef{\gls@hypergroup@list@#1}%
  {\protected@csxappto{\gls@hypergroup@list@#1}{, #2}}%
  {\protected@csxdef{\gls@hypergroup@list@#1}{#2}}%
}
\newcommand*{\glsnavigation}{%
  \def\@gls@between{}%
  \ifcsundef{\gls@hypergroup@list@\currentglossary}%
  {%
    \def\@gls@list{}%
  }%
  {%
    \letcs\@gls@list{\gls@hypergroup@list@\currentglossary}%
  }%
  \@for\@gls@tmp:=\@gls@list\do{%
    \@gls@between
  }
```

```

\@gls@getgrouptitle{\@gls@tmp}{\@gls@grptitle}%
\expandafter\glsnavhyperlink\expandafter{\@gls@tmp}{\@gls@grptitle}%
\let\@gls@between\glshypernavsep
}%
}
\newcommand*{\glshypernavsep}{\space\textbar\space}
\newcommand*{\glssymbolnav}{%
\glsnavhyperlink{glssymbols}{\glsgetgrouptitle{glssymbols}}%
\glshypernavsep
\glsnavhyperlink{glsnumbers}{\glsgetgrouptitle{glsnumbers}}%
\glshypernavsep
}

```

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}%
    }%
  \fi
}

```

```

}%
\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]{%
  \beginngroup
  \protected@edef\@gls@tmp{#1}%
  \@onelevel@sanitize\@gls@tmp
  \expandafter\glsshowtarget\expandafter{\@gls@tmp}%
  \endgroup
}
\newcommand*{\@glsshowaccsupp}[3]{%
\newrobustcmd*{\glsshowaccsupp}[3]{%
  \ifstrempy{#1}%
  {\@@glsshowtarget{/#2 (#3)}}%
  {\@@glsshowtarget{/#2 (#3) [#1]}}%
}
}

```



```

\newcommand*{\@gls@see@noindex}{%
  \PackageError{glossaries}%
  {'\gls@xr@key' key may only be used after \string\makeglossaries\space
  or \string\makenoidxglossaries\space (or move
  \string\newglossaryentry\space
  definitions into the preamble)}}%
{You must use \string\makeglossaries\space
  or \string\makenoidxglossaries\space before defining
  any entries that have a '\gls@xr@key' key. It may
  be that the 'see' key has been written to the .glsdefs
  file from the previous run, in which case you need to
  move your definitions
  to the preamble if you don't want to use
  \string\makeglossaries\space
  or \string\makenoidxglossaries}%
}
\define@choicekey{glossaries.sty}{seenoinindex}{%
  [\gls@seenoinindex@val\gls@seenoinindex@nr]{error,warn,ignore}{%
    \ifcase\gls@seenoinindex@nr
      \renewcommand*{\@gls@see@noindex}{%
        \PackageError{glossaries}%
        {'\gls@xr@key' key may only be used after \string\makeglossaries\space
        or \string\makenoidxglossaries}%
        {You must use \string\makeglossaries\space
        or \string\makenoidxglossaries\space before defining
        any entries that have a '\gls@xr@key' key}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{%
        \GlossariesWarning{'\gls@xr@key' key ignored}%
      }%
    \or
      \renewcommand*{\@gls@see@noindex}{}%
    \fi
  }
\define@boolkey{glossaries.sty}[gls]{toc}[true]{}
\define@boolkey{glossaries.sty}[gls]{numberline}[true]{}
\ifcsundef{chapter}%
  {\newcommand*{\@@glossarysec}{section}}%
  {\newcommand*{\@@glossarysec}{chapter}}
\define@choicekey{glossaries.sty}{section}{part,chapter,section,%
subsection,subsubsection,paragraph,subparagraph}[section]{%
  \renewcommand*{\@@glossarysec}{#1}}
\newcommand*{\@@glossarysecstar}{*}
\newcommand*{\@@glossaryseclabel}{}
\newcommand*{\glsautoprefix}{}
\define@choicekey{glossaries.sty}{numberedsection}%
  [\gls@numberedsection@val\gls@numberedsection@nr]{%
    false,nolabel,autolabel,nameref}[nolabel]{%
      \ifcase\gls@numberedsection@nr\relax

```

```

\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{%
\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\@ifpackageloaded{classicthesis}
{\newcommand*{\@glossary@default@style}{index}}
{\newcommand*{\@glossary@default@style}{list}}
\define@key{glossaries.sty}{style}{%
\def\@glossary@default@style{#1}%
}
\newcommand*{\@gls@declareoption}[2]{%
\DeclareOptionX{#1}{#2}%
\DeclareOption{#1}{#2}%
}
\newcommand*{\glossaryentrynumbers}[1]{#1\gls@save@numberlist{#1}}
\@gls@declareoption{nonumberlist}{%
\renewcommand*{\glossaryentrynumbers}[1]{\gls@save@numberlist{#1}}%
}
\define@boolkey{glossaries.sty}[gls]{savenumberlist}[true]{}
\gls@savenumberlistfalse
\newcommand*{\@glo@seeautonumberlist}{}
\@gls@declareoption{seeautonumberlist}{%
\renewcommand*{\@glo@seeautonumberlist}{%
\def\@glo@prefix{\glsnextpages}%
}%
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{}
\gls@esclocationsfalse
\newcommand*{\@gls@loadlong}{\RequirePackage{glossary-long}}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}%
}
\IfFileExists{supertabular.sty}{%
\newcommand*{\@gls@loadsuper}{\RequirePackage{glossary-super}}}%
\newcommand*{\@gls@loadsuper}{}%
\@ifpackageloaded{xtab}{\renewcommand*{\@gls@loadsuper}{}%
}
\@gls@declareoption{nosuper}{\renewcommand*{\@gls@loadsuper}{}%
}
\newcommand*{\@gls@loadlist}{\RequirePackage{glossary-list}}
\@gls@declareoption{nolist}{%
\renewcommand*{\@gls@loadlist}{%
}

```

```

\ifdefstring{\@glossary@default@style}{list}%
{\let\@glossary@default@style\relax}%
}%
}%
}
\newcommand*{\@gls@loadtree}{\RequirePackage{glossary-tree}}
\@gls@declareoption{notree}{\renewcommand*{\@gls@loadtree}{}}
\@gls@declareoption{nostyles}{%
\renewcommand*{\@gls@loadlong}{}%
\renewcommand*{\@gls@loadsuper}{}%
\renewcommand*{\@gls@loadlist}{}%
\renewcommand*{\@gls@loadtree}{}%
\let\@glossary@default@style\relax
}
\newcommand*{\glspostdescription}{%
\ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{}
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{}
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{}
\@ifclassloaded{memoir}
{%
\glsucmarktrue
}%
{%
\glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
\ifglsentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%
\else
\newcounter{glossaryentry}[\@gls@counterwithin]%
\fi
\def\theHglossaryentry{\currentglossary.\theglossaryentry}%
}%
}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{}
\glsentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
\renewcommand*{\@gls@counterwithin}{#1}%
\glsentrycountertrue
\@gls@define@glossaryentrycounter
}

```

```

\newcommand*{\@gls@counterwithin}{%
\newcommand{\@gls@define@glossarysubentrycounter}{%
  \ifundef\c@glossarysubentry
  {%
    \ifglssubentrycounter
    \ifglsubentrycounter
      \newcounter{glossarysubentry}[glossaryentry]%
    \else
      \newcounter{glossarysubentry}%
    \fi
    \def\theH@glossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
  \fi
}%
}%
}
\define@boolkey{glossaries.sty}{gls}{subentrycounter}[true]{%
\glssubentrycounterfalse
\newcommand{\@gls@indexingsetting}{\ifglsubentry 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*{\@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*\gls@sortnumberfmt[1]{%
  \ifnum#1<100000 0\fi
  \ifnum#1<10000 0\fi
  \ifnum#1<1000 0\fi
  \ifnum#1<100 0\fi
  \ifnum#1<10 0\fi
  \number#1%
}

```

```

\newcommand*{\@gls@setupsort@def}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
  }%
  \def\@gls@defsort##1##2{%
    \ifcsundef{glossary@##1@sortcount}%
    {\@gls@defsortcount{##1}}%
    {}%
    \expandafter\global\expandafter
    \advance\csname glossary@##1@sortcount\endcsname by 1\relax
    \expandafter\protected@xdef\csname glo@##2@sort\endcsname{%
      \expandafter\glssortnumberfmt
      {\csname glossary@##1@sortcount\endcsname}}%
  }%
  \def\@gls@setsort##1{%
    \let\@glo@check@sortallowed\@gobble
  }
\newcommand*{\@gls@setupsort@use}{%
  \let\do@glo@storeentry\@gobble
  \def\@gls@defsortcount##1{%
    \expandafter\global
    \expandafter\newcount\csname glossary@##1@sortcount\endcsname
  }%
  \def\@gls@defsort##1##2{%
    \expandafter\gdef\csname glo@##2@sort\endcsname{%
  }%
  \def\@gls@setsort##1{%
    \protected@edef\@glo@parent{\csname glo@##1@parent\endcsname}%
    \ifx\@glo@parent\@empty
    \else
      \expandafter\@gls@setsort\expandafter{\@glo@parent}%
    \fi
    \protected@edef\@glo@type{\csname glo@##1@type\endcsname}%
    \protected@edef\@gls@tmp{\csname glo@##1@sort\endcsname}%
    \ifx\@gls@tmp\@empty
      \expandafter\global\expandafter
      \advance\csname glossary@\@glo@type @sortcount\endcsname by 1\relax
      \expandafter\protected@xdef\csname glo@##1@sort\endcsname{%
        \expandafter\glssortnumberfmt
        {\csname glossary@\@glo@type @sortcount\endcsname}}%
      \@glo@storeentry{##1}%
    \fi
  }%
  \let\@glo@check@sortallowed\@gobble
}
\newcommand*{\@gls@setupsort@none}{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%

```

```

\def\@gls@defsortcount##1{%
\def\@gls@defsort##1##2{%
  \expandafter\global\expandafter\let\csname glo@##2@sort\endcsname\@glo@sort
}%
\def\@gls@setsort##1{%
\renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
{Option sort=none not allowed with \string##1}%
{(Use sort=def instead)}}%
}
\newcommand*\@gls@setupsort@clear{%
  \def\glsindexingsetting{none}%
  \def\do@glo@storeentry##1{%
    \def\@gls@defsortcount##1{%
      \def\@gls@defsort##1##2{\csgdef{glo@##2@sort}{}}%
      \def\@gls@setsort##1{%
        \renewcommand\@glo@check@sortallowed[1]{\PackageError{glossaries}
        {Option sort=clear not allowed with \string##1}%
        {(Use sort=def instead)}}%
      }%
    }%
  }
\newcommand*\@glsdefmain{%
  \if@gls@docloaded
    \newglossary[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*\@glsacronymtype{\@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*\@glsacronymtype{acronym}%
      \newcommand*\@gls@tr@set@acronym@toctitle{%
        \translatelet{\glossarytoctitle}{Acronyms}%
      }%
    }%
  \else
    \let\@gls@do@acronymsdef\relax
  \fi
}
\AtBeginDocument{%

```

```

\ifglsacronym
  \providecommand*\printacronyms[1][1]{%
    \printglossary[type=\acronymtype,#1]}%
\fi
}
\newcommand*\@gls@do@acronymsdef{}
\@gls@declareoption{acronyms}{%
  \glsacronymtrue
  \def\@gls@do@acronymsdef{%
    \DeclareAcronymList{acronym}%
    \newglossary[alg]{acronym}{acr}{acn}{\acronymname}%
    \renewcommand*\@acronymtype{acronym}%
    \newcommand*\@gls@tr@set@acronym@toctitle{%
      \translatelet{\glossarytoctitle}{Acronyms}%
    }%
  }%
}
\newcommand*\@glsacronymlists{}
\newcommand*\@addtoacronymlists[1]{%
  \ifx\@glsacronymlists\@empty
    \protected@xdef\@glsacronymlists{#1}%
  \else
    \protected@xdef\@glsacronymlists{\@glsacronymlists,#1}%
  \fi
}
\newcommand*\@DeclareAcronymList[1]{%
  \glsIfListOfAcronyms{#1}{\@addtoacronymlists{#1}}%
}
\newcommand*\@glsIfListOfAcronyms[1]{%
  \protected@edef\@do@gls@islistofacronyms{%
    \noexpand\@gls@islistofacronyms{#1}{\@glsacronymlists}}%
  \@do@gls@islistofacronyms
}
\newcommand*\@gls@islistofacronyms[4]{%
  \def\@gls@islistofacronyms##1,#1,##2\end@gls@islistofacronyms{%
    \def\@gls@before{##1}\def\@gls@after{##2}}%
  \gls@islistofacronyms,#2,#1,\@nil\end@gls@islistofacronyms
  \ifx\@gls@after\@nnil
    #4%
  \else
    #3%
  \fi
}
\newif\if@gls@isacronymlist
\newcommand*\@gls@checkisacronymlist[1]{%
  \glsIfListOfAcronyms{#1}%
  {\@gls@isacronymlisttrue}{\@gls@isacronymlistfalse}%
}
\newcommand*\@SetAcronymLists[1]{%
  \renewcommand*\@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*{\glssetexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@expand@field{##1}{#1}{##2}%
  }%
}
\newcommand*{\glssetnoexpandfield}[1]{%
  \csdef{gls@assign@#1@field}##1##2{%
    \@gls@noexpand@field{##1}{#1}{##2}%
  }%
}
\glssetexpandfield{type}
\glssetnoexpandfield{desc}
\glssetnoexpandfield{descplural}
\newcommand*{\@gls@sanitizename}{}
\glssetnoexpandfield{name}
\newcommand*{\@gls@sanitizesymbol}{}
\glssetnoexpandfield{symbol}
\glssetnoexpandfield{symbolplural}
\newcommand*{\@gls@sanitizesort}{%
  \ifglssanitizesort
    \@gls@sanitizesort
  \else
    \@gls@nosanitizesort
  \fi
}
\newcommand*{\@gls@sanitizesort}{%
  \@onelevel@sanitize\@glo@sort
}
\newcommand*{\@gls@nosanitizesort}{}
\newcommand*{\@gls@noidx@sanitizesort}{%
  \ifdefvoid\@glo@sort
    {}%
  {%
    \expandafter\@gls@noidx@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
}
\def\@gls@noidx@sanitizesort#1#2\gls@end@sanitizesort{%
  \def\@glo@sort{#1#2}%
  \@onelevel@sanitize\@glo@sort
}

```

```

}
\newcommand*{\@@gls@noidx@nosanitizesort}{%
  \ifdefvoid\@glo@sort
  {%
  {%
    \expandafter\@@gls@noidx@no@sanitizesort\@glo@sort\gls@end@sanitizesort
  }%
  }
}
\def\@@gls@noidx@no@sanitizesort#1#2\gls@end@sanitizesort{%
  \bgroup
  \glsnoidxstripaccents
  \protected@xdef\@@glo@sort{#1#2}%
  \egroup
  \let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
  \let\IeC\@firstofone
  \let\add@accent@\@secondoftwo
  \let\@text@composite@x\@secondoftwo
  \let\@tabacckludge\@secondoftwo
  \expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
  \expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
  \let\''\@firstofone
  \let\''\@firstofone
  \let\~\@firstofone
  \let\""\@firstofone
  \let\u\@firstofone
  \let\t\@firstofone
  \let\d\@firstofone
  \let\r\@firstofone
  \let\=\@firstofone
  \let\.\@firstofone
  \let\~\@firstofone
  \let\v\@firstofone
  \let\H\@firstofone
  \let\c\@firstofone
  \let\b\@firstofone
  \let\a\@secondoftwo
  \def\AE{AE}%
  \def\ae{ae}%
  \def\OE{OE}%
  \def\oe{oe}%
  \def\AA{AA}%
  \def\aa{aa}%
  \def\L{L}%
  \def\l{l}%
  \def\O{O}%
  \def\o{o}%

```

```

\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r\fmtversion{2019/10/01}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[gls]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifgls@sanitize@symbol
\glssetnoexpandfield{symbol}%
\glssetnoexpandfield{symbolplural}%
\else
\glssetexpandfield{symbol}%
\glssetexpandfield{symbolplural}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
\ifglssanitizesort
\glssetnoexpandfield{sortvalue}%
\renewcommand*{\@gls@noidx@setsanitizesort}{%
\glssanitizesorttrue

```

```

        \glsssetnoexpandfield{sortvalue}%
    }%
\else
    \glsssetexpandfield{sortvalue}%
    \renewcommand*{\@gls@noidx@setsanitizesort}{%
        \glsssanitizesortfalse
        \glsssetexpandfield{sortvalue}%
    }%
\fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}{true,false}[true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }%
    \setkeys[gls]{sanitize}{#1}%
}%
}
\newif\ifglstranslate
\newcommand*{\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
        \let\glsifusetranslator\@secondoftwo
    }%
    {%
        \@ifpackageloaded{babel}%
        {%
            \IfFileExists{translator.sty}%

```

```

        {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
        }%
        {}%
    }%
    {}%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
    \glsifusetranslator
    {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
    {#3}%
}
\@gls@declareoption{nottranslate}{%
    \glstranslatefalse
    \let\@gls@usetranslator\relax
    \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}%
[\gls@translate@val\gls@translate@nr]%
{true,false,babel}[true]%
{%
    \ifcase\gls@translate@nr\relax
    \glstranslatetrue
    \renewcommand*\@gls@usetranslator{%
        \@ifpackageloaded{polyglossia}%
        {%
            \let\glsifusetranslator\@secondoftwo
        }%
        {%
            \@ifpackageloaded{babel}%
            {%
                \IfFileExists{translator.sty}%
                {%
                    \RequirePackage{translator}%
                    \let\glsifusetranslator\@firstoftwo
                }%
                {}%
            }%
            {}%
        }%
    }%
}
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax

```

```

        \let\glsifusetranslator\@secondoftwo
    \fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
    \glstranslatetrue
    \let\glsifusetranslator\@firstoftwo
}%
{%
    \@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
    {
        \@ifpackageloaded{\gls@thissty}%
        {%
            \glstranslatetrue
            \@endfortrue
        }%
    }%
}
}
\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{\language}%
\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}}
\ifdefined\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}{%
  \ifdequal\makeglossaries\@no@makeglossaries
    {%
      \GlossariesWarning{Option 'disablemakegloss' has no effect
        (\string\makenoidxglossaries\space has already been used)}%
    }
  }

```

```

}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option 'disablemakegloss' has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\renewcommand{\@domakeglossaries}[1]{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been disabled}%
}%
}%
}%
}
\@gls@declareoption{restoremakegloss}{%
\ifdefequal\makeglossaries\@no@makeglossaries
{%
\GlossariesWarning{Option 'restoremakegloss' has no effect
(\string\makenoidxglossaries\space has already been used)}%
}%
{%
\ifdefequal\@makeglossary\@gobble
{%
\GlossariesWarning{Option 'restoremakegloss' has no effect
(\string\makeglossaries\space has already been used)}%
}%
{%
\PackageInfo{glossaries}{\string\makeglossaries\space and
\string\makenoidxglossaries\space have been restored}%
\let\@domakeglossaries\@firstofone
}%
}%
}
}
\newcommand*{\@do@write@glslabels}{%
\AtEndDocument{\@do@write@glslabels}%
\let\@do@write@glslabels\relax
}
\newcommand*{\@do@write@glslabels}{%
\newwrite\@glsl@labelsfile
\immediate\openout\@glsl@labelsfile=\jobname.glslabels
\protected@edef\@glsl@fulllist{\@gls@types}%
\ifdefempty{\@ignored@glossaries}{%
{\protected@eappto\@glsl@fulllist{\@ignored@glossaries}}%
\forallgl@sentries[\@glsl@fulllist]{\@glsl@entry}%
}%
\ifdefempty{\@glsl@entry}{%
{%
\ifx\do@write@glslabelsname\@gobble
\def\@glsl@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 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}{\glssymbolsgroupname}%
  \newcommand*{\printsymbols}[1][\@gls@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{\@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]
\ifdefined\@gls@requested@languages{%
{%
  \@for\@gls@thislang:=\@gls@requested@languages\do{%
    \TrackIfKnownLanguage{\@gls@thislang}{}%
    {\GlossariesWarning{unknown language ‘\@gls@thislang’}}%
  }%
}%
\@gls@usetranslator
\@ifpackageloaded{translator}
{%
  \ifboolexpr
  {
    test {\ifdefstring{\trans@languages}{English}}
    and not
    test {\ifdefstring{bbl@loaded}{english}}
  }
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \usedictionary{glossaries-dictionary}%
    \renewcommand*{\addglossarytocaptions}[1]{%
      \ifcsundef{captions#1}{}%
      {%
        \expandafter\let\expandafter\@gls@tmp\csname captions#1\endcsname
        \expandafter\toks@\expandafter{\@gls@tmp
          \renewcommand*{\glossaryname}{\translate{Glossary}}}%
        }%
        \expandafter\edef\csname captions#1\endcsname{\the\toks@}%
      }%
    }%
  }%
}%
\AnyTrackedLanguages
{%
  \ForEachTrackedDialect{\this@dialect}{%
    \IfTrackedLanguageFileExists{\this@dialect}%
    {glossaries-}% prefix
    {.ldf}%
    {%
      \RequireGlossariesLang{\CurrentTrackedTag}%
    }%
    {%
      \@gls@missinglang@warn\this@dialect\CurrentTrackedLanguage
    }%
  }%
}%
}%

```

```

\glsifusetranslator
{%
  \renewcommand*{\glssettoctitle}[1]{%
    \ifcsdef{gls@tr@set@#1@toctitle}%
    {%
      \csuse{gls@tr@set@#1@toctitle}%
    }%
    {%
      \def\glossarytoctitle{\csname @glotype@#1@title\endcsname}%
    }%
  }%
  \renewcommand*{\glossaryname}{\translate{Glossary}}%
  \renewcommand*{\acronymname}{\translate{Acronyms}}%
  \renewcommand*{\entryname}{\translate{Notation (glossaries)}}%
  \renewcommand*{\descriptionname}{%
    \translate{Description (glossaries)}}%
  \renewcommand*{\symbolname}{\translate{Symbol (glossaries)}}%
  \renewcommand*{\pagelistname}{%
    \translate{Page List (glossaries)}}%
  \renewcommand*{\glssymbolsgroupname}{%
    \translate{Symbols (glossaries)}}%
  \renewcommand*{\glsnumbersgroupname}{%
    \translate{Numbers (glossaries)}}%
  }{%
\fi
\DeclareRobustCommand*{\nopostdesc}{%
\newcommand*{\@nopostdesc}{%
  \let\org@glspostdescription\glspostdescription
  \def\glspostdescription{%
    \let\glspostdescription\org@glspostdescription}%
}
\newcommand*{\@no@post@desc}{\nopostdesc}
\newcommand{\glspar}{\par}
\newcommand{\setStyleFile}[1]{%
  \renewcommand*{\gls@istfilebase}{#1}%
  \ifglsxindy
    \def\istfilename{\gls@istfilebase.xdy}
  \else
    \def\istfilename{\gls@istfilebase.ist}
  \fi
}
\@onlypremakeg\setStyleFile
\ifglsxindy
  \def\istfilename{\gls@istfilebase.xdy}
\else
  \def\istfilename{\gls@istfilebase.ist}
\fi
\newcommand*{\gls@istfilebase}{\jobname}
\newcommand*{\@istfilename}[1]{%
\newcommand*{\glscompositor}{.}

```



```

\newcommand*\glsSetCompositor}[1]{%
  \renewcommand*\glscompositor{#1}}
\@onlypremakeg\glsSetCompositor
\newcommand*\@glsAlphacompositor{\glscompositor}
\ifglxindy
  \newcommand*\glsSetAlphaCompositor[1]{%
    \renewcommand*\@glsAlphacompositor{#1}}
\else
  \newcommand*\glsSetAlphaCompositor[1]{%
    \glsnxindywarning\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{\glslocationncstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*\glossarypreamble{%
  \csuse{@glossarypreamble@\currentglossary}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*\glossarypostamble{}
\newcommand*\glossarysection[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls glossarymark{\glossarytoctitle}%
}

```

```

\ifcsundef{glossarymark}%
{%
  \newcommand{\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{\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*{\@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*\@gls@clearpage{\@gls@docclearpage}
\newcommand*\@gls@toc[2]{%
\ifglstoc
\ifglslnumberline
\addcontentsline{toc}{#2}{\protect\numberline{#1}%
\else
\addcontentsline{toc}{#2}{#1}%
\fi
\fi
}
\newcommand*\@gls@noxindywarning[1]{%
\GlossariesWarning{Not in xindy mode --- ignoring \string#1}%
}
\newcommand*\@gls@nomakeindexwarning[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\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@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,#2,#1,\end@doifinlist
}
\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]{%
    \glsnoindywarning\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*\@glsaddxdyattribute[2]{%
    \edef\xdyattributes{\@xdyattributes ^^J
      \string"#2#1\string"%
    \expandafter\toks@\expandafter{\@xdylocref}%
    \edef\xdylocref{\the\toks@ ^^J%
      (markup-locref

```

```

:open \string"\glstildechar n%
\expandafter\string\csname glsX#2X#1\endcsname
\string" ^^J
:close \string"\string" ^^J
:attr \string"#2#1\string"}}%
\expandafter\gdef\csname glsX#2X#1\endcsname##1##2{%
\setentrycounter[##1]{#2}\csname #1\endcsname{##2}%
}%
}
\newcommand*\GlsAddXdyAttribute[1]{%
\ifx\@xdyattributelist\@empty
\edef\@xdyattributelist{#1}%
\else
\edef\@xdyattributelist{\@xdyattributelist,#1}%
\fi
\edef\@xdyattributes{\@xdyattributes ^^J
\string"#1\string"%
\@for\@this@counter:=\@xdycounters\do{%
\protected\edef\gls@do@addxdyattribute{%
\noexpand\@glsaddxdyattribute{#1}{\@this@counter}%
}
\gls@do@addxdyattribute
}%
\let\GlsAddXdyCounters\@disabled@glsaddxdycounters
}
\@onlypremakeg\GlsAddXdyAttribute
\else
\newcommand*\GlsAddXdyAttribute[1]{%
\glsnoxywarning\GlsAddXdyAttribute}
\fi
\ifglsxindy
\newcommand*{\@gls@addpredefinedattributes}{%
\GlsAddXdyAttribute{glsnumberformat}
\GlsAddXdyAttribute{textrm}
\GlsAddXdyAttribute{textsf}
\GlsAddXdyAttribute{texttt}
\GlsAddXdyAttribute{textbf}
\GlsAddXdyAttribute{textmd}
\GlsAddXdyAttribute{textit}
\GlsAddXdyAttribute{textup}
\GlsAddXdyAttribute{textsl}
\GlsAddXdyAttribute{textsc}
\GlsAddXdyAttribute{emph}
\GlsAddXdyAttribute{glshypernumber}
\GlsAddXdyAttribute{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{
\ifglxsindy
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\edef\@xdyuseralphabets{%
\@xdyuseralphabets ^^J
(define-alphabet "#1" (#2))}}
\else
\newcommand*{\GlsAddXdyAlphabet}[2]{%
\glsnoxywarning\GlsAddXdyAlphabet}
\fi
\ifglxsindy
\newcommand*{\@gls@xdy@locationlist}{%
roman-page-numbers,%
Roman-page-numbers,%
arabic-page-numbers,%
alpha-page-numbers,%
Alpha-page-numbers,%
Appendix-page-numbers,%
arabic-section-numbers%
}
\protected@edef\@gls@roman{\@roman{0}\string"
\string"roman-numbers-lowercase\string" :sep \string"}}%
\@onelevel@sanitize\@gls@roman
\edef\@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]{%
  \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
\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
  \ifglxindy@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*{\forallsentries}[3][\glsdefaulttype]{%
  \protected@edef\@glo@list{\csname glolist@#1\endcsname}%
  \@for#2:=\@glo@list\do
  {%
    \ifdefempty{#2}{\#{#3}%
  }%
}
\newcommand*{\forallglsentries}[3][\@glo@types]{%
  \expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
  {%
    \forallsentries[\@this@glo@]{#2}{#3}%
  }%
}
\newcommand{\ifglossaryexists}{%
  \@ifstar\s@ifglossaryexists\@ifglossaryexists
}
\newcommand{\@ifglossaryexists}[3]{%
  \ifcsundef{glo@type@#1@out}{#3}{#2}%
}
\newcommand{\s@ifglossaryexists}[3]{%
  \ifcsundef{glolist@#1}{#3}{#2}%
}
\newcommand*{\glsdetoklabel}[1]{#1}
\newcommand{\ifglsentryexists}[3]{%
  \ifcsundef{glo@\glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*{\ifglsused}[3]{%
  \ifbool{glo@\glsdetoklabel{#1}@flag}{#2}{#3}%
}
\newcommand{\glsdoifexists}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}{You need to define a glossary entry before you
      can use it.}}%
}
\newcommand{\glsdoifnoexists}[2]{%
  \ifglsentryexists{#1}{%
    \PackageError{glossaries}{Glossary entry ‘\glsdetoklabel{#1}’ has already
      been defined}{\#2}%
}
\newcommand{\glsdoifexistsorwarn}[2]{%
  \ifglsentryexists{#1}{#2}{%
    \GlossariesWarning{Glossary entry ‘\glsdetoklabel{#1}’
      has not been defined}%
  }%
}

```

```

}
\newcommand{\glsdoifexistsordo}[3]{%
  \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\forglsentries\expandafter
      [\csname glo@\@gls@thislabel @type\endcsname]
    {\glo@label}%
    {%
      \letcs\glo@parent{glo@\glo@label @parent}%
      \ifdefequal\@gls@thislabel\glo@parent
      {%
        \def\do@glshaschildren{#2}%
        \@endfortrue
      }%
    }%
  }%
  \do@glshaschildren
}%
}
\newcommand{\ifglshasparent}[3]{%
  \glsdoifexists{#1}%
  {%
    \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@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}%
    }%
  }
}
\newcommand{\ifglfieldvoid}[4]{\ifcsvoid{glo@glstetoklabel{#2}@#1}{#3}{#4}}
\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*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
    \protected@edef\@glo@tmp{#1}%
    \ifx\@glo@tmp\@gls@defaulttype
        \GlossariesWarning{\string\defglsentryfmt\space attempting to set
            a display style for \string\glsdefaulttype\space but there
            doesn’t seem to be a default glossary. If you have used
            ‘nomain’, you need to define an alternative glossary}%
    \else
        \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
    \fi
}

```

```

\newcommand*\gls@doentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*\@gls@forbidtexext}[1]{%
\ifbool{test {\ifdefstring{#1}{tex}}
or test {\ifdefstring{#1}{TEX}}}%
{%
\def#1{nottex}%
\PackageError{glossaries}%
{Forbidden '.tex' extension replaced with '.nottex'}%
{I'm sorry, I can't allow you to do something so reckless.\MessageBreak
Don't use '.tex' as an extension for a temporary file.}%
}%
}%
}%
}
\newcommand*\gls@gobbleopt{\new@ifnextchar[\@gls@gobbleopt]}
\def\@gls@gobbleopt[#1]{%
\newcommand*\@newglossary{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*\s@newglossary}[2]{%
\ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*\ns@newglossary}[5][glg]{%
\doifglossarynoexistsordo{#2}%
{%
\ifundef\glsdefaultttype
{%
\gdef\glsdefaultttype{#2}%
}%}%
\toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
\expandafter\gdef\csname glolist@#2\endcsname{,}%
\expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
\expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
\expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
\expandafter\@gls@forbidtexext\csname @glotype@#2@log\endcsname
\expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
\expandafter\@gls@forbidtexext\csname @glotype@#2@out\endcsname
\expandafter\def\csname @glotype@#2@title\endcsname{#5}%
\@gls@provide@newglossary
\protected@write\@auxout{\string\@newglossary{#2}{#1}{#3}{#4}}%
\ifcsundef{gls@#2@entryfmt}%
{%
\defglsentryfmt[#2]{\glsentryfmt}%
}%
}%
\@gls@defsortcount{#2}%
\@ifnextchar[\@gls@setcounter{#2}]{%
{\@gls@setcounter{#2}[\glscounter]}%
}%
}%
\gls@gobbleopt
}%

```

```

}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%
}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{}
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @glotype@#1@counter\endcsname{#2}%
  \ifglxsindy
    \GlsAddXdYCounters{#2}%
  \fi
}
\newcommand*{\@gls@getcounter}[1]{%
  \csname @glotype@#1@counter\endcsname
}
\glsdefmain
\@gls@do@acronymsdef
\@gls@do@symbolsdef
\@gls@do@numbersdef
\@gls@do@indexdef
\newcommand*{\newignoredglossary}[1]{%
  \ifdefempty\@ignored@glossaries
  {%
    \protected@edef\@ignored@glossaries{#1}%
  }%
  {%
    \protected@eappto\@ignored@glossaries{, #1}%
  }%
  \csgdef{glolist@#1}{,}%
  \ifcsundef{gls@#1@entryfmt}%
  {%
    \defglsentryfmt[#1]{\glsentryfmt}%
  }%
  {}%
  \ifdefempty\@gls@nohyperlist
  {%
    \renewcommand*{\@gls@nohyperlist}{#1}%
  }%
  {%
    \protected@eappto\@gls@nohyperlist{, #1}%
  }%
}
\newcommand*{\@ignored@glossaries}{}
\newcommand*{\ifignoredglossary}[3]{%
  \protected@edef\@gls@igtype{#1}%
  \expandafter\DTLifinlist\expandafter
  {\@gls@igtype}{\@ignored@glossaries}{#2}{#3}%
}
\define@key{glossentry}{name}{%

```

```

\def\@glo@name{#1}%
}
\define@key{glossentry}{description}{%
\def\@glo@desc{#1}%
}
\define@key{glossentry}{descriptionplural}{%
\def\@glo@descplural{#1}%
}
\define@key{glossentry}{sort}{%
\def\@glo@sort{#1}}
\define@key{glossentry}{text}{%
\def\@glo@text{#1}%
}
\define@key{glossentry}{plural}{%
\def\@glo@plural{#1}%
}
\define@key{glossentry}{first}{%
\def\@glo@first{#1}%
}
\define@key{glossentry}{firstplural}{%
\def\@glo@firstplural{#1}%
}
\newcommand*{\@gls@default@value}{\relax}
\define@key{glossentry}{symbol}{%
\def\@glo@symbol{#1}%
}
\define@key{glossentry}{symbolplural}{%
\def\@glo@symbolplural{#1}%
}
\define@key{glossentry}{type}{%
\def\@glo@type{#1}}
\define@key{glossentry}{counter}{%
\ifcsundef{c@#1}%
{%
\PackageError{glossaries}%
{There is no counter called ‘#1’}%
{%
The counter key should have the name of a valid counter
as its value%
}%
}%
{%
\def\@glo@counter{#1}%
}%
}
\define@key{glossentry}{see}{%
\gls@set@xr@key{see}{\@glo@see}{#1}%
}
\newcommand*{\gls@set@xr@key}[3]{%
\renewcommand*{\gls@xr@key}{#1}%

```



```

\gls@checkseeallowed
\def#2{#3}%
\@glo@seeautonumberlist
}
\newcommand*{\gls@xr@key}{see}
\newcommand*{\gls@checkseeallowed}{%
\@gls@see@noindex
}
\newcommand*{\gls@checkseeallowed@preambleonly}{%
\GlossariesWarning
{'\gls@xr@key' key doesn't have any effect when used in the document
environment. Move the definition to the preamble
after \string\makeglossaries\space
or \string\makenoidxglossaries}%
}
\define@key{glossentry}{parent}{%
\def\@glo@parent{#1}}
\define@choicekey{glossentry}{nonumberlist}%
[\gls@nonumberlist@val\gls@nonumberlist@nr]{true,false}[true]%
{%
\ifcase\gls@nonumberlist@nr\relax
\def\@glo@prefix{\glsnonextpages}%
\@gls@savenonumberlist{true}%
\else
\def\@glo@prefix{\glsnextpages}%
\@gls@savenonumberlist{false}%
\fi
}
\newcommand*{\gls@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*{\@glsnodelsc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'}.

```

```

    Maybe makeglossaries + rerun required}%
\else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \edef\@gls@value{\expandonce{#1}}%
    \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
  \edef\@gls@value{\expandonce{#1}}%
  \@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
  \@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
  \protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \edef\@gls@value{\expandonce{#1}}%
    \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
  }%

```

```

}%
{%
  \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}%
}%
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \@@gls@expand@field{#2}{#3}{#1}%
  }%
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\gls@defdocnewglossaryentry}{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}

```

```

\newcommand*{\gls@glsdefs@labellist}{%
\newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*{\gls@docdef@labellist}{%
\newcommand*{\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
\ifundef\@gls@deffile
{%
\global\newwrite\@gls@deffile
\immediate\openout\@gls@deffile=\jobname.glsdefs
}%
}%
\ifglsentryexists{#1}%
{%
\xifinlist{#1}{\gls@docdef@labellist}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in the
document environment (ignoring new definition)}%
}%
{%
\xifinlist{#1}{\gls@glsdefs@labellist}%
{%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in
the preamble (in which case remove the duplicate definition)
or you have just updated to glossaries.sty
v4.47 (in which case delete ‘\jobname.glsdefs’)}%
}%
}%
}%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*{\gls@begindocdefs}{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode‘\noexpand\@=\number\catcode‘\@‘\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{%}{%
\@gls@restoreat
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}

```

```

\newcommand*{\glsdefs@newdocentry}[2]{%
  \ifglentryexists{#1}%
    {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
      (if you have moved the definition there from the document
        environment, delete \jobname.glsdefs)}}%
    {%
      \gls@glsdefs@addlabel{#1}%
      \gls@defglossaryentry{#1}{#2}%
    }%
}
\newcommand*{\@gls@writedef}[1]{%
  \immediate\write\@gls@deffile
  {%
    \string\glsdefs@newdocentry{\glsdetoklabel{#1}}\glspercentchar^^J%
    \expandafter\@gobble\string\{\glspercentchar%
  }%
  \@for\@gls@map:=\@gls@keymap\do
  {%
    \letcs\glo@value{\glo@\glsdetoklabel{#1}\expandafter\@secondoftwo\@gls@map}%
    \ifdef\glo@value
    {%
      \@onelevel@sanitize\glo@value
      \immediate\write\@gls@deffile
      {%
        \expandafter\@firstoftwo\@gls@map
        =\expandafter\@gobble\string\{\glo@value\expandafter\@gobble\string\},%
        \glspercentchar
      }%
    }%
  }%
}
\gls.writedefhook
\immediate\write\@gls@deffile
{%
  \glspercentchar^^J%
  \expandafter\@gobble\string\}\glspercentchar^^J%
}%
}
\newcommand*{\@gls@keymap}{%
  {name}{name},%
  {sort}{sortvalue},% unescaped sort value
  {type}{type},%
  {first}{first},%
  {firstplural}{firstpl},%
  {text}{text},%
  {plural}{plural},%
  {description}{desc},%
  {descriptionplural}{descplural},%
  {symbol}{symbol},%
  {symbolplural}{symbolplural},%

```

```

{user1}{useri},%
{user2}{userii},%
{user3}{useriii},%
{user4}{useriv},%
{user5}{userv},%
{user6}{uservi},%
{long}{long},%
{longplural}{longpl},%
{short}{short},%
{shortplural}{shortpl},%
{counter}{counter},%
{parent}{parent}%
}
\newcommand*{\@gls@fetchfield}[2]{%
  \edef\@gls@thisval{#2}%
  \@for\@gls@map:=\@gls@keymap\do{%
    \edef\@this@key{\expandafter\@firstoftwo\@gls@map}%
    \ifdefequal{\@this@key}{\@gls@thisval}%
    {%
      \edef#1{\expandafter\@secondoftwo\@gls@map}%
      \@endfortrue
    }%
  }%
}%
}
\newcommand*{\glsaddstoragekey}{\@ifstar\@sglsaddstoragekey\@glsaddstoragekey}
\newcommand*{\@sglsaddstoragekey}[1]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \expandafter\newcommand\expandafter*\expandafter
    {\csname gls@assign@#1@field\endcsname}[2]{%
      \@gls@expand@field{##1}{#1}{##2}%
    }%
  }%
  {}%
  \@glsaddstoragekey{#1}%
}
\newcommand*{\@glsaddstoragekey}[3]{%
  \key@ifundefined{glossentry}{#1}%
  {%
    \define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
    \appto\@gls@keymap{, {#1}{#1}}%
    \appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
    \appto\@newglossaryentryposthook{%
      \letcs{@glo@tmp}{@glo@#1}%
      \gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
    }%
    \newcommand*{#3}[1]{\@gls@entry@field{##1}{#1}}%
  }%
  {}%
}

```

```

\PackageError{glossaries}{Key ‘#1’ already exists}{}%
}%
}
\newcommand*{\glsaddkey}{\@ifstar\sglsaddkey\@glsaddkey}
\newcommand*{\@sglsaddkey}[1]{%
\key@ifundefined{glossentry}{#1}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname gls@assign@#1@field\endcsname}[2]{%
\@Gls@expand@field{##1}{#1}{##2}%
}%
}%
}%
}%
\@glsaddkey{#1}%
}
\newcommand*{\@Glsaddkey}[7]{%
\key@ifundefined{glossentry}{#1}%
{%
\define@key{glossentry}{#1}{\csdef{@glo@#1}{##1}}%
\appto\@gls@keymap{,{#1}{#1}}%
\appto\@newglossaryentryprehook{\csdef{@glo@#1}{#2}}%
\appto\@newglossaryentryposthook{%
\letcs{@glo@tmp}{@glo@#1}%
\gls@assign@field{#2}{\@glo@label}{#1}{\@glo@tmp}%
}%
\newcommand*{#3}[1]{\@Gls@entry@field{##1}{#1}}%
\newcommand*{#4}[1]{\@Gls@entry@field{##1}{#1}}%
\ifcsdef{@gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#5’ as helper command
‘\expandafter\string\csname @gls@user@#1@\endcsname’ already exists}%
}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
{\csname @gls@user@#1\endcsname}[2][{}]{%
\new@ifnextchar[%
{\csuse{@gls@user@#1@}{##1}{##2}}%
{\csuse{@gls@user@#1@}{##1}{##2}[{}]}%
\csdef{@gls@user@#1@}##1##2[##3]{%
\@gls@field@link{##1}{##2}{#3{##2}##3}%
}%
\newrobustcmd*{#5}{%
\expandafter\@gls@hyp@opt\csname @gls@user@#1\endcsname}%
}%
\ifcsdef{@Gls@user@#1@}%
{%
\PackageError{glossaries}%
{Can’t define ‘\string#6’ as helper command

```



```

        '\expandafter\string\csname @Gls@user@#1@\endcsname' already exists}%
    }%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @Gls@user@#1@\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@Gls@user@#1@}{##1}{##2}}%
      {\csuse{@Gls@user@#1@}{##1}{##2} []}}%
\csdef{@Gls@user@#1@}##1##2[##3] {%
  \gls@field@link{##1}{##2}{#4{##2}##3}%
}%
\newrobustcmd*{#6}{%
  \expandafter\@gls@hyp@opt\csname @Gls@user@#1@\endcsname}%
}%
\ifcsdef{@GLS@user@#1@}%
{%
  \PackageError{glossaries}%
  {Can't define '\string#7' as helper command
   '\expandafter\string\csname @GLS@user@#1@\endcsname' already exists}%
  {}%
}%
{%
\expandafter\newcommand\expandafter*\expandafter
  {\csname @GLS@user@#1@\endcsname}[2] [] {%
    \new@ifnextchar [%
      {\csuse{@GLS@user@#1@}{##1}{##2}}%
      {\csuse{@GLS@user@#1@}{##1}{##2} []}}%
\csdef{@GLS@user@#1@}##1##2[##3] {%
  \@gls@field@link{##1}{##2}{\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{\glsfieldxddef}[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@csdef{glo@\@glo@label @#2}{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldgdef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\gdef\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%

```

```

        \letcs#3{glo@\@glo@label @#2}%
    }%
    {%
        \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
}%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
    \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
        \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
    \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\ifglsfielddcseq}[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]{%
  \ifglstryexists{#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@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{}%
  }%
\fi

```

```

}%
{%
\ifglentryexists{\@glo@parent}%
{%
\expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
\@glo@parent}%
\gls@level=\csname glo@\@glo@parent @level\endcsname\relax
\advance\gls@level by 1\relax
\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
{%
\PackageError{glossaries}%
{%
Invalid parent '\@glo@parent'
for entry '\@glo@label' - parent doesn't exist%
}%
{%
Parent entries must be defined before their children%
}%
\def\@glo@parent{}%
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
{\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname}%
{\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @plural\endcsname}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\else
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\fi

```

```

\ifcsundef{@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
  \let{@glo@name}@gls@name
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@{@glo@label @numberlist}}%
{%
  \csxdef{glo@{@glo@label @numberlist}}{%
    \noexpand@gls@missingnumberlist{@glo@label}}%
}%
{%
  \let{@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@desc
  \let{@glo@desc}@gls@desc
\fi
\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\@gl@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{\@gl@detoklabel{#1}}%
\protected@edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
\def\@glo@list{,%
\for@gl@entries[\@glo@type]{\@glo@label}%
{%
\ifdefequal\@glo@thislabel\@glo@label
}{\protected@eappto\@glo@list{\@glo@label,}}%
}%
\cslet\@glolist{\@glo@type}{\@glo@list}%
\csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifgl@xindy
\newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else

```



```

\newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglxindy
\newcommand*{\@glossarysubentryfield}{%
\string\subglossentry}
\else
\newcommand*{\@glossarysubentryfield}{%
\string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
\protected@edef\@glo@esclabel{#1}%
\@gls@checkmkidxchars\@glo@esclabel
\protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
\@gls@checkmkidxchars\@glo@sort
\@gls@checkmkidxchars\@glo@prefix
\protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%
\ifglxindy
\ifx\@glo@parent\@empty
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
(\string"\@glo@sort\string" %
\string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname
(\string"\@glo@sort\string" %
\string"\@glo@prefix\@glossarysubentryfield
{\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
}%
\fi
\else
\ifx\@glo@parent\@empty
\@onelevel@sanitize\@glo@prefix
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossaryentryfield{\@glo@esclabel}%
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}{\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*{\@gls@if@measuring}[2]{%
\ifmeasuring@
#1%

```

```

\else
  #2%
\fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand{\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
  \ifpackageloaded{amsmath}%
  {\let\gls@if@measuring\@gls@if@measuring}%
  }%
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter[1]{%
  \ifcsdef{c#1}%
  {%
    \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
    \expandafter\advance\csname c#1\endcsname by \@ne
  }%
  }%
}
\newcommand{\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo
  \let\@glslink\@secondoftwo
}
\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{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\{glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\{glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\{glsunset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsunset{#1}%
}%
}%
}
}

```

```

\newcommand*{\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \glslocalunset{#1}%
    }%
  }%
}
\newcommand*{\@glslocalunset}{\@glslocalunset}
\newcommand*{\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*{\@glsunset}{\@glsunset}
\newcommand*{\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*{\@glslocalreset}{\@glslocalreset}
\newcommand*{\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*{\@glsreset}{\@glsreset}
\newcommand*{\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*{\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*{\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*{\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*{\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*{\@newglossaryentry@defcounters}{%

```

```

\csdef{glo@%glo@label @currcount}{0}%
\csdef{glo@%glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*{\glsenableentrycount}{%
\let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
\renewcommand*{\gls@defdocnewglossaryentry}{%
\renewcommand*newglossaryentry[2]{%
\PackageError{glossaries}{\string\newglossaryentry\space
may only be used in the preamble when entry counting has
been activated}{If you use \string\glsenableentrycount\space
you must place all entry definitions in the preamble not in
the document environment}%
}%
}%
\newcommand*{\glsentrycurrcount}[1]{%
\ifcsundef{glo@%glsdetoklabel{##1}@currcount}%
{0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
\ifcsundef{glo@%glsdetoklabel{##1}@prevcount}%
{0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
\@@glsunset{##1}%
\@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
\@@glslocalunset{##1}%
\@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
\@@glsreset{##1}%
\ifglsresetcurrcount\csgdef{glo@%glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*{\@glslocalreset}[1]{%
\@@glslocalreset{##1}%
\ifglsresetcurrcount\csdef{glo@%glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgls@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglsformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}[##3]%
\fi
}%
\def\@cglspl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\@cGls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \cGls@{##1}{##2}[##3]%
    \fi
}%
\def\@cGlspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cGlspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@cglsetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@cglsetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@cglsetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*\@string\@gls@entry@count}[2]{}}%
    \forallglsetentries{\@glsetentry}{%
        \ifglsetused{\@glsetentry}%
        {\immediate\write\@auxout
            {\string\@gls@entry@count{\@glsetentry}{\glsetentrycurrcount{\@glsetentry}}}}%
        {}%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
}
\newrobustcmd*{\cglset}{\@gls@hyp@opt\@cglset}
\newcommand*{\@cglset}[2][{}]{%

```

```

\new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
\GlossariesWarning{\string\cgl\space is defaulting to
\string\gl\space since you haven't enabled entry counting}%
\@cgl@{#1}{#2}[#3]%
}
\newcommand*{\cgl@format}[2]{%
\ifglshaslong{#1}{\gl@entrylong{#1}}{\gl@entryfirst{#1}}#2%
}
\newrobustcmd*{\cgl@}{\@cgl@hyp@opt\@cgl@}
\glsmfuaddmap{\cgl@}{\cgl@}
\newcommand*{\@cgl@}[2][]{%
\new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
\GlossariesWarning{\string\cgl\space is defaulting to
\string\gl\space since you haven't enabled entry counting}%
\@cgl@{#1}{#2}[#3]%
}
\newcommand*{\cgl@format}[2]{%
\ifglshaslong{#1}{\gl@entrylong{#1}}{\gl@entryfirst{#1}}#2%
}
\newrobustcmd*{\cgl@pl}{\@cgl@hyp@opt\@cgl@pl}
\glsmfuaddmap{\cgl@pl}{\cgl@pl}
\newcommand*{\@cgl@pl}[2][]{%
\new@ifnextchar[{\@cgl@pl@{#1}{#2}}{\@cgl@pl@{#1}{#2}[]}%
}
\def\@cgl@pl@#1#2[#3]{%
\GlossariesWarning{\string\cgl@pl\space is defaulting to
\string\gl@pl\space since you haven't enabled entry counting}%
\@cgl@pl@{#1}{#2}[#3]%
}
\newcommand*{\cgl@pl@format}[2]{%
\ifglshaslong{#1}{\gl@entrylongpl{#1}}{\gl@entryfirstplural{#1}}#2%
}
\newrobustcmd*{\cgl@pl}{\@cgl@hyp@opt\@cgl@pl}
\glsmfuaddmap{\cgl@pl}{\cgl@pl}
\newcommand*{\@cgl@pl}[2][]{%
\new@ifnextchar[{\@cgl@pl@{#1}{#2}}{\@cgl@pl@{#1}{#2}[]}%
}
\def\@cgl@pl@#1#2[#3]{%
\GlossariesWarning{\string\cgl@pl\space is defaulting to
\string\gl@pl\space since you haven't enabled entry counting}%
\@cgl@pl@{#1}{#2}[#3]%
}
\newcommand*{\cgl@pl@format}[2]{%
\ifglshaslong{#1}{\gl@entrylongpl{#1}}{\gl@entryfirstplural{#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
{%
\glsupercase{\glsentryplural{\glslabel}\glsinsert}%
}%
{%
\glsupercase{\glsentryfirstplural{\glslabel}\glsinsert}%
}%
}%
}%
\glscapscase
{%
\ifglsused\glslabel
{%
\glsentrytext{\glslabel}\glsinsert
}%
{%
\glsentryfirst{\glslabel}\glsinsert
}%
}%
}

```



```

}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \glsuppercase{\Glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \glsuppercase{\Glsentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \glsuppercase
            {\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
      }%
      {%

```

```

        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \glsuppercase
        {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \glsuppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
    {%
        \glscapscase
        {%
            \genacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \glsuppercase
            {\genacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
}%
{%
    \glscustomtext
}%
}
\newcommand*{\genacrfullformat}[2]{%
    \glsentrylong{#1}#2\space
    (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
    \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
    \expandafter\glssentencecase\expandafter{\gls@text}%
}

```

```

\newcommand*{\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*{\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*{\@gls@glslink@hyper@update@hook}{%
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef{\hyperlink}{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*{\@gls@save@glslocal}{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*{\@gls@restore@glslocal}{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset\glslocalunset
  \else
    \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*{\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{%
\@gls@save@glslocal
\newcommand*{\glslinkvar}[3]{#1}
\newcommand*{\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*{\s@gls@hyp@opt}[1][{}]{%
  \let\glslinkvar\@secondofthree

```

```

\@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][{}]{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
\@gls@hyp@opt\@gls@@link
}
\newcommand*{\@gls@@link}[3][{}]{%
\glsdoifexistsordo{#2}%
{%
\let\do@gls@link@checkfirsthyper\relax
\@gls@link[#1]{#2}{#3}%
}%
\glstextformat{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@@link
}
\newcommand*{\@Gls@@link}[3][{}]{%
\@gls@@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
\ifglsused{\glslabel}%
{%
}%
{%
\gls@checkisacronymlist\glstype
\ifglshyperfirst
\if@glsisacronymlist
\ifglsacrfootnote
\KV@glslink@hyperfalse
\fi
\fi
\else
\KV@glslink@hyperfalse
\fi
}%
\glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
\expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
{\KV@glslink@hyperfalse}{}%
}

```

```

\newcommand*{\@gls@setdefault@glslink@opts}{%
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper@ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \fi
  \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
  \@gls@restore@glslocal
}
\newcommand*{\glsencapwrcontent}[1]{#{#1}}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
\def\@gls@Hcounter{%
\ifthenelse{equal{\@gls@counter}{equation}}%
{
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglsentrycounter}%

```

```

    }%
    {%
        \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
    }%
    \protected@edef\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@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}%
      }%
      \gls@dostubst
    }%
  \fi
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifgls@xindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescactual#1\@nil\?\?\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkactual#1\@nil??\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkbar#1\@nil||\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescbar#1\@nil\\|\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \fi
}

```

```

\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
\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
\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@quotearchar\@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@quotearchar\@gls@levelchar\@gls@quotearchar\@gls@levelchar}%
            \def\@gls@checklevel{\@gls@checklevel#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotearchar\@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@quotearchar\@gls@actualchar\@gls@quotearchar\@gls@actualchar}%
            \def\@gls@checkactual{\@gls@checkactual#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotearchar\@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
}
\bggroup
\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}%
\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:nnT
    { \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{\gl:disablehyper}{%
  \KV@glslink@hyperfalse
  \let\@glslink\glndonohyperlink
  \let\@glstarget\@secondoftwo
}
\newcommand{\gl:enablehyper}{%
  \KV@glslink@hypertrue
  \let\@glslink\glsdohyperlink
  \let\@glstarget\glsdohypertarget
}
\providecommand{\@firstofthree}[3]{#1}
\providecommand{\@secondofthree}[3]{#2}
\newrobustcmd*{\gls}{\@glshyp@opt\@gls}
\newcommand*{\@gls}[2][{}]{%
  \new@ifnextchar[{\@gls@{#1}{#2}}{\@gls@{#1}{#2}[]}]%
}
\def\@gls@#1#2[#3]{%
  \gl:doifexists{#2}%
  {%
    \let\do@glslink@checkfirsthyper\@glslink@checkfirsthyper
    \let\gl:sifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\gl:insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@glslink[#1]{#2}{\@glo@text}%
    \@gl:do@gl:sunset{#2}%
  }%
  \gl:postlinkhook
}
\newrobustcmd*{\Gls}{\@glshyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}]%
}
\def\@Gls@#1#2[#3]{%
  \gl:doifexists{#2}%
  {%
    \let\do@glslink@checkfirsthyper\@glslink@checkfirsthyper
    \let\gl:sifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\gl:insert{#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@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\gl@sifplural\@firstoftwo
  \let\gl@scapscase\@secondofthree
  \let\gl@customtext\@empty
  \def\gl@insert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \@gl@do@gl@unset{#2}%
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLSpl}{\@gl@hyp@opt\@GLSpl}
\gl@smfublocker{\GLSpl}
\newcommand*{\@GLSpl}[2][{}]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@doifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gl@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
  \@gl@disp[#1]{#2}{\gl@sentencecase{#3}}%
}
\gl@smfuaddmap{\gl@disp}{\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}{\glstentrytext{#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}{\glssupercase{\glstentrytext{#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}{\Glstentrytext{#2}{#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][]{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\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}{\glssupercase{\glsentryfirst{#2}{#3}}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}

```

```

\newcommand*{\@glsplural}[2][\%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2}[]}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2][\%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLsplural}{\@gls@hyp@opt\@GLsplural}
\glsmfublocker{\GLsplural}
\newcommand*{\@GLsplural}[2][\%
  \new@ifnextchar[{\@GLsplural@{#1}{#2}}{\@GLsplural@{#1}{#2}[]}]
\def\@GLsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][\%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][\%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\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*{\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}{\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}{\glssupercase{\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}{\glssupercase{\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}{\glssupercase{\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}{\glssupercase{\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}{\glsupercase{\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}{\glsupercase{\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}{\glssupercase{\glssentryuservi{#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]{%
\glstdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@firstofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\glssentryshort{#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]{%
\glstdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@secondoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\acronymfont{\Glsentryshort{#2}}#3%
}%
\@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\@ns@ACRshort}[2] [] {%

```



```

\new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\gl@ifplural\@secondoftwo
\let\gl@scapscase\@thirdofthree
\let\gl@insert\@empty
\def\glscustomtext{%
\glssupercase{\acronymfont{\gl@entryshort{#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\gl@scapscase\@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}
\glsmfuaddmap{\Acrshortpl}{\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\gl@scapscase\@secondofthree
\let\gl@insert\@empty
\def\glscustomtext{%

```

```

        \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][{}]{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2}[]}%
}
\def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\glscapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \glsupercase{\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\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlong}{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glssupercase{\Glsentrylong{#2}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\acrlongpl}{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
  }%
}

```

```

    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][{}]{%
  \new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \Glsentrylongpl{#2}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][{}]{%
  \new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glssupercase{\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{\@Gls@entry@field}[2]{%
  \glstexpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
\ifdef\texpdfstring
{\newcommand{\glstexpdfstring}{\texpdfstring}}
{\newcommand{\glstexpdfstring}[2]{#1}}
\newcommand{\@Gls@entryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Gls@entryname}[1]{%
  \glstexpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{name}}}%
}
\glsmfuexcl{\Gls@entryname}
\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\Gls@entrylong\relax

```

```

\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\glssentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*{\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Glsentryfield{#1}{desc}}
\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*{\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}%
}
\glsmfuexcl{\Glsentryuseri}
\newcommand*{\glsentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
}
\newcommand*{\Glsentryuserii}[1]{%
    \@Glsentryfield{#1}{userii}%
}
\glsmfuexcl{\Glsentryuserii}
\newcommand*{\glsentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
}
\newcommand*{\Glsentryuseriii}[1]{%
    \@Glsentryfield{#1}{useriii}%
}
\glsmfuexcl{\Glsentryuseriii}
\newcommand*{\glsentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
}
\newcommand*{\Glsentryuseriv}[1]{%
    \@Glsentryfield{#1}{useriv}%
}
\glsmfuexcl{\Glsentryuseriv}
\newcommand*{\glsentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
}
\newcommand*{\Glsentryuserv}[1]{%
    \@Glsentryfield{#1}{userv}%
}
\glsmfuexcl{\Glsentryuserv}
\newcommand*{\glsentryuservi}[1]{%
    \@gls@entry@field{#1}{uservi}%
}
\newcommand*{\Glsentryuservi}[1]{%
    \@Glsentryfield{#1}{uservi}%
}
\glsmfuexcl{\Glsentryuservi}
\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*{\glentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuexcl{\Glsentrylong}
\newcommand*{\glentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuexcl{\Glsentrylongpl}
\newcommand*{\glentryfull}[1]{%
  \glentrylong{#1} (\glentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glentryshort{#1})%
}
\glsmfuexcl{\Glsentryfull}
\newcommand*{\glentryfullpl}[1]{%
  \acrfullformat{\glentrylongpl{#1}}{\acronymfont{\glentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glentryshortpl{#1})%
}
\glsmfuexcl{\Glsentryfullpl}
\newcommand*{\glentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glentrynumberlist\space instead%
    }%
    \glentrynumberlist{#1}%
  }%
}%
{\%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup
      \protected@edef\glo@label{\glsdetoklabel{#1}}%
      \let\@org@glnumberformat\glnumberformat
      \def\glnumberformat##1{##1}%
    }
  }
}

```

```

\protected@edef\the@numberlist{%
  \csname glo@\glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{%
\def\@gls@numlist@nextsep{%
\def\@gls@numlist@lastsep{%
\def\@gls@thislist{%
\def\@gls@donext@def{%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\gls hyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\glo linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\glsmfuexcl{\glsadd}
\newcommand*{\@gls@adjustmode}{%
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*{\glsaddall}[1][\@glo@types]{%
  \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*{\@glsmminrange}{2}
\ifglsxindy
\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
\renewcommand*{\@glsmminrange}{#1}}
\else
\newcommand*{\GlsSetXdyMinRangeLength}[1]{%
\glsnoxywarning\GlsSetXdyMinRangeLength}
\fi

\ifglsxindy
\def\writeist{%
\ifundef{\glswrite}{\newwrite\glswrite}{}%
\@glscataddpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@glsc@classI:=\@glsc@xdy@locationlist\do{%
\protected@write\glswrite{}{(define-location-class
\string"\@glsc@classI\string"^^J\space\space\space
(
:sep "{ }{"
\csname @glsc@xdy@Lclass@\@glsc@classI\endcsname\space
:sep "}"
)
^^J\space\space\space

```

```

:min-range-length \@glsminrange^^J%
)
}%
{%
\@for\@gls@classII:=\@gls@xdy@locationlist\do{%
\protected@write\glswrite{}\{(define-location-class
\string"\@gls@classII-\@gls@classI\string"
^^J\space\space\space
(
:sep "{"
\csname @gls@xdy@Lclass@\@gls@classII\endcsname\space
:sep "{{"
\csname @gls@xdy@Lclass@\@gls@classI\endcsname\space
:sep "}"
)
^^J\space\space\space
:min-range-length \@glsminrange^^J%
)
}%
}%
}%
}%
\write\glswrite{^^J; user defined location classes}%
\write\glswrite{\@xdyuserlocationdefs}%
\write\glswrite{^^J; define cross-reference class^^J}%
\write\glswrite{(define-crossref-class \string"see\string"
:unverified )}%
\write\glswrite{(markup-crossref-list
:class \string"see\string"^^J\space\space\space
:open \string"\string\glsseeformat\string"
:close \string"{}\string")}%
\@xdycrossrefhook
\write\glswrite{^^J; define the order of the location classes}%
\write\glswrite{(define-location-class-order
(\@xdylocationclassorder))}%
\write\glswrite{^^J; define the glossary markup^^J}%
\write\glswrite{(markup-index^^J\space\space\space
:open \string"\string
\glossarysection[\string\glossarytoctitle]{\string
\glossarytitle}\string\glossarypreamble}%
\@for\@this@ctr:=\@xdycounters\do{%
{%
\@for\@this@attr:=\@xdyattributelist\do{%
\protected@write\glswrite{}\{\string\providecommand*%
\expandafter\string
\csname glsX\@this@ctr X\@this@attr\endcsname[2]%
{%
\string\setentrycounter
[\expandafter\@gobble\string\#1]{\@this@ctr}%
\expandafter\string

```

```

\csname\@this@attr\endcsname
{\expandafter\@gobble\string\#2}%
}%
}%
}%
}%
\write\glswrite{%
\string\begin
{theglossary}\string\glossaryheader\glstildechar n\string" ^^J\space
\space\space:close \string"\glpercentchar\glstildechar n\string
\end{theglossary}\string\glossarypostamble
\glstildechar n\string" ^^J\space\space\space
:tree)}}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{(markup-indexentry
:open \string"\string\relax \string\glresetentrylist
\glstildechar n\string"}}%
\write\glswrite{(markup-locclass-list :open
\string"\glsoopenbrace\string\glossaryentrynumbers
\glsoopenbrace\string\relax\space \string"^^J\space\space\space
:sep \string", \string"
:close \string"\glsclosebrace\glsclosebrace\string"}}%
\write\glswrite{(markup-locref-list
:sep \string"\string\delimN\space\string"}}%
\write\glswrite{(markup-range
:sep \string"\string\delimR\space\string"}}%
\@onelevel@sanitize\gls@suffixF
\@onelevel@sanitize\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixF" :length 1 :ignore-end)}}%
\fi
\ifx\gls@suffixFF\@empty
\else
\write\glswrite{(markup-range
:close "\gls@suffixFF" :length 2 :ignore-end)}}%
\fi
\write\glswrite{^^J; define format to use for locations^^J}%
\write\glswrite{\@xdylocref}%
\write\glswrite{^^J; define letter group list format^^J}%
\write\glswrite{(markup-letter-group-list
:sep \string"\string\glsgroupskip\glstildechar n\string"}}%
\write\glswrite{^^J; letter group headings^^J}%
\write\glswrite{(markup-letter-group
:open-head \string"\string\glsgroupheading
\glsoopenbrace\string"^^J\space\space\space
:close-head \string"\glsclosebrace\string"}}%

```

```

\write\glswrite{^^J; additional letter groups^^J}%
\write\glswrite{\@xdylettergroups}%
\write\glswrite{^^J; additional sort rules^^J}
\write\glswrite{\@xdysortrules}%
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\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{\glspcrcentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspcrcentchar\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}
\write\glswrite{delim_n \string}\string\delimN \string}
\write\glswrite{delim_r \string}\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string}\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
    \string}\string}\string\relax
    \string\glsresetentrylist \string}
\write\glswrite{symhead_positive \string}glssymbols\string}
\write\glswrite{numhead_positive \string}glsnumbers\string}
\write\glswrite{page_compositor \string}\glscompositor\string}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
    \write\glswrite{suffix_2p \string}\gls@suffixF\string}
\fi
\ifx\gls@suffixFF\@empty
\else
    \write\glswrite{suffix_3p \string}\gls@suffixFF\string}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\let\noist\@gls@noist@toolate
}
\fi
\newcommand*\GlsSetWriteIstHook}[1]{\renewcommand*\@gls@writeisthook}{#1}}
\@onlypremakeg\GlsSetWriteIstHook
\newcommand*\@gls@writeisthook}{}
\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####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@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%

```

```

\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesbar####1\noexpand\@nil
\noexpand||\noexpand||\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\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
\noexpand\@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\@gls@openfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%

```

```

\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@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{-l }}%
{\let\@gls@order\@empty}%
\edef\@gls@dothiswrite{\noexpand\write18{makeindex \@gls@order

```

```

        -s \istfilename\space
        -t \jobname.\csuse{@glotype@#1@log}
        -o \jobname.\csuse{@glotype@#1@in}
        \jobname.\csuse{@glotype@#1@out}}}%
    }%
    \@gls@dothiswrite
}%
{%
    \GlossariesWarning{Can't make glossary '#1', it doesn't exist}%
}%
}
\fi
\ifglxindy
\newcommand*{\@gls@automake@immediate}[1]{%
    \ifglossaryexists{#1}
    {%
        \IfFileExists{\jobname.\csuse{@glotype@#1@out}}}%
        {%
            \ifdefstring{\glsorder}{letter}%
            {\def\@gls@order{-M ord/letorder }}%
            {\let\@gls@order\@empty}%
            \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@type@#1@log}
-o \jobname.\csuse{\gls@type@#1@in}
\jobname.\csuse{\gls@type@#1@out}}}%
}%
\@gls@dothiswrite
}%
{\GlossariesWarning{can't automake '#1': \jobname.\csuse{\gls@type@#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@onlypreamble\makeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries\warn@noprintglossary
\iflssavenumberlist
  \edef\@gls@dodeflistparser{\noexpand\DeclareListParser
    {\noexpand\glsnumlistparser}{\delimN}}%
  \@gls@dodeflistparser
\fi
\let\makenoidxglossaries\@no@makeglossaries
\renewcommand*\@printgloss@setsort{%
  \let\@glo@assign@sortkey\@glo@no@assign@sortkey
}%
\ifglsautomake
  \renewcommand*\@gls@doautomake{%
    \protected@edef\gls@tmp{\@gls@automake@types}%
    \@for\@gls@type:=\gls@tmp\do{%
      \ifdefempty{\@gls@type}{}%
        {\@gls@automake{\@gls@type}}%
      }%
    }%
  \fi
  \@glo@check@sortallowed\makeglossaries
}%
}
\@onlypreamble{\makeglossaries}

```

```

\newcommand\@makeglossaries@warn@noprintglossary{%
  \ifdefstring{\@glo@types}{,}%
  {%
    \GlossariesWarningNoLine{No glossaries have been defined}%
  }%
  {%
    \GlossariesWarningNoLine{No \string\printglossary\space
      or \string\printglossaries\space
      found. ^^J(Remove \string\makeglossaries\space if you
      don't want any glossaries.) ^^JThis document will not
      have a glossary}%
  }%
}%
\AtEndDocument{%
  \warn@nomakeglossaries
  \warn@noprintglossary
}
\newcommand*\@makenoidxglossaries{%
  \def\glsindexingsetting{noidx}%
  \@domakeglossaries
  {%
    \renewcommand{\@gls@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\@gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
          {\csname glo@\@gls@label @type\endcsname}%
          {\@gls@label}%
        {%
          \string\glsseeformat##2}%
        }%
      }%
    }%
  }%
\AtBeginDocument
{%
  \write\@auxout{\string\providecommand\string\@gls@reference[3]{}%
}%
\def\warn@noprintglossary{%
  \GlossariesWarningNoLine{No \string\printnoidxglossary\space
    or \string\printnoidxglossaries ^^J

```



```

        found. (Remove \string\makenoidxglossaries\space if you
        don't want any glossaries.)^^JThis document will not have a glossary}%
}%
\let\warn@nomakeglossaries\relax
\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
    \let\@glo@assign@sortkey\@glo@assign@sortkey
    \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
    \PackageError{glossaries}{Glossary entries must be
        defined in the preamble^^Jwhen you use
        \string\makenoidxglossaries}%
    {Either move your definitions to the preamble or use
        \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
    \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
    \ifdef\@gls@loclist
    {%
        \glsnoidxloclist{\@gls@loclist}%
    }%
    {%
        ??\glsdoifexists{##1}%
        {%
            \GlossariesWarning{Missing location list for '##1'. Either
                a rerun is required or you haven't referenced the entry}%
        }%
    }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
    \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
    \ifdef\@gls@loclist
    {%
        \def\@gls@noidxloclist@sep{%
            \def\@gls@noidxloclist@sep{%
                \def\@gls@noidxloclist@sep{%
                    \glsnumlistsep
                }%
                \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
            }%
            \def\@gls@noidxloclist@finalsep{}%
            \def\@gls@noidxloclist@prev{}%
            \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
            \@gls@noidxloclist@finalsep
            \@gls@noidxloclist@prev
        }%
        {%
            ??\glsdoifexists{##1}%

```

```

    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@glstetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc\glsnoidxdisplayloc
  \let\@gls@org@glssseeformat\glssseeformat
  \let\glsnoidxdisplayloc##2\relax
  \let\glssseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let\glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let\glssseeformat\@gls@org@glssseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}

```

```

}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\auxout{\glswrglossdisablelocationcmds}{%
    \string\@gls@reference
    {\csname glo@\@gls@label @type\endcsname}%
    {\@gls@label}%
    {\string\glsnoidxdisplayloc
      {\@glo@counterprefix}%
      {\@gls@counter}%
      {\@glsnumberformat}%
      {\@glslocref}}%
    }%
  }%
}
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%
        \expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
      \immediate\closeout\glswrite
    \fi
  }%
}
\newcommand*{\gls@glossary}[1]{%
  \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%

```

```

\if@gls@debug
  \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
\fi
}
\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begingroup\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
    \expandafter{\@gls@tmp^^J}%
  \else
    \ifcsdef{glo@#1@file}%
    {%
      \expandafter\protected@write\csname glo@#1@file\endcsname{%
        \gls@disablepagerefexpansion}{#2}%
      }%
    {%
      \ifignoredglossary{#1}{}%
      {%
        \GlossariesWarning{No file defined for glossary ‘#1’}%
      }%
    }%
  \fi
  \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*{\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
\glswrglossdisablelocationcmds
\ifglsclocations
  \@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]{%
  \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\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{}%
  \fi
}

```

```

\else
  \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
\protected@edef\@gls@label{\@glsdetoklabel{#1}}%
\@@do@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@do@esc@wrglossary}[1]{% please read documented code!
  \begingroup
    \let\gls@orgthe\the
    \let\gls@orgnumber\number
    \let\gls@orgarabic\@arabic
    \let\gls@orgromannumeral\romannumeral
    \let\gls@orgalph\@alph
    \let\gls@orgAlph\@Alph
    \let\gls@orgRoman\@Roman
    \ifglswrallowprimitivemods
      \def\gls@the##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
      \def\the{\expandafter\gls@the}%
      \def\gls@number##1{%
        \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
      \def\number{\expandafter\gls@number}%
      \def\romannumeral##1{%
        \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
    \fi
    \def\@arabic##1{%
      \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
    \def\@Roman##1{%
      \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
    \def\@alph##1{%
      \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
    \def\@Alph##1{%
      \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
    \@wrglossarynumberhook
    \gls@disablepagerefexpansion
    \protected@xdef\@glslocref{\theHglentrycounter}%
  \endgroup
  \@gls@checkmkidxchars\@glslocref
  \expandafter\ifx\theHglentrycounter\theHglentrycounter\relax
  \def\@glo@counterprefix{}%
\else
  \protected@edef\@glsHlocref{\theHglentrycounter}%
  \@gls@checkmkidxchars\@glsHlocref
  \xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
\fi
\protected@edef\@gls@label{\@glsdetoklabel{#1}}%
\@@do@wrglossary
}

```

```

\newcommand*{\@do@wrglossary}{%
  \ifglxsindy
    \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
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    (indexentry :tkey (\csname glo@\@gls@label @index\endcsname)
      :locref \string"\@glo@counterprefix{\@glslocref}\string" %
      :attr \string"\@gls@counter\@glo@suffix\string"
      \@glo@range
    )
  }%
\else
  \@set@glo@numformat{\@glo@numfmt}{\@gls@counter}{\@glsnumberformat}%
  {\@glo@counterprefix}%
  \gls@glossary{\csname glo@\@gls@label @type\endcsname}{%
    \string@glossaryentry{\csname glo@\@gls@label @index\endcsname
      \@gls@encapchar\@glo@numfmt}{\@glslocref}}%
  \fi
}
\newcommand*\@xp@gls@getcounterprefix[2]{%
  \bgroup
  \glswrglossdisableanchorcmds
  \protected@edef\@do@gls@getcounterprefix{%
    \noexpand\egroup
    \noexpand\@gls@getcounterprefix{#1}{#2}%
  }%
  \@do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
  \def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
  \ifx\@gls@thisloc\@gls@thisHloc
    \def\@glo@counterprefix{}%
  \else
    \def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
      \def\@glo@tmp{##2}%
      \ifx\@glo@tmp\@empty
        \def\@glo@counterprefix{}%
      \else
        \def\@glo@counterprefix{##1}%
      \fi
    }%
    \@gls@get@counterprefix#2.#1\end@getprefix
    \ifx\@glo@counterprefix\@empty

```

```

\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
  prefixing^^Jlocation ‘#1’. You need to modify the
  definition of \string\theH\@gls@counter^^Jotherwise you
  will get the warning: “‘name{\@gls@counter.#1}’ has been^^J
  referenced but does not exist"%
\fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglxsindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
  (indexentry
  :tkey (\csname glo@#1@index\endcsname)
  :xref (\string"\@gls@xref\string")
  :attr \string"see\string"
  )
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*\@glssee}[3][\seename]{%
\@do@seeglossary{#2}{[#1]{#3}}}
\newcommand*\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*\@glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}
\newrobustcmd*\@glsseelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext

```



```

\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseelastsep
\let\@gls@donext\glsseesep
}%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }
\newrobustcmd*{\glsseeitem}[1]{\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}{%
  \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{\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@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{\glsetoklabel{##2}}%
\gls@org@glossarysubentryfield{##1}{##2}%
}%
\@gls@preglossaryhook
#2%
}%
\gls@printglossary@end
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprntglossary\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}%
\ifglxindy
\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@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}{@glsref@\@glo@type}%
    {%
    }%
    {%
    }%
  }%
}

```

```

        \expandafter\@glo@do@sortentries\expandafter{\@glo@parent}%
    }%
}%
{%
    \@glo@sortedinsert{\@glo@sortinglist}{#1}%
}%
}
\newcommand*{\@glo@sortedinsert}[2]{%
    \dtl@insertinto{#2}{#1}{\@glo@sortinghandler}%
}%
\newcommand*{\@glo@sorthandler@word}[2]{%
    \letcs\@gls@sort@A{\@glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{\@glo@\glsdetoklabel{#2}@sort}%
    \edef\@glo@do@compare{%
        \noexpand\dtlwordindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \@glo@do@compare
}
\newcommand*{\@glo@sorthandler@letter}[2]{%
    \letcs\@gls@sort@A{\@glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{\@glo@\glsdetoklabel{#2}@sort}%
    \edef\@glo@do@compare{%
        \noexpand\dtlletterindexcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \@glo@do@compare
}
\newcommand*{\@glo@sorthandler@case}[2]{%
    \letcs\@gls@sort@A{\@glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{\@glo@\glsdetoklabel{#2}@sort}%
    \edef\@glo@do@compare{%
        \noexpand\dtlcompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \@glo@do@compare
}
\newcommand*{\@glo@sorthandler@nocase}[2]{%
    \letcs\@gls@sort@A{\@glo@\glsdetoklabel{#1}@sort}%
    \letcs\@gls@sort@B{\@glo@\glsdetoklabel{#2}@sort}%
    \edef\@glo@do@compare{%
        \noexpand\dtlicompare{\noexpand\dtl@sortresult}%
        {\expandonce\@gls@sort@B}%
        {\expandonce\@gls@sort@A}%
    }%
    \@glo@do@compare
}

```

```

\newcommand*{\@glo@sortmacro@word}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@word}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=word]}{}%
  }%
}

\newcommand*{\@glo@sortmacro@letter}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@letter}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=letter]}{}%
  }%
}

\newcommand*{\@glo@sortmacro@standard}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \ifcsdef{\@glo@sorthandler@glorder}%
    {%
      \@glo@sortentries{\csuse{\@glo@sorthandler@glorder}}{#1}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\glorder’}{}%
    }%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=standard]}{}%
  }%
}

\newcommand*{\@glo@sortmacro@case}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@case}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=case]}{}%
  }%
}

```

```

\newcommand*{\@glo@sortmacro@nocase}[1]{%
  \ifdefstring{\@glo@default@sorttype}{standard}%
  {%
    \@glo@sortentries{\@glo@sorthandler@nocase}{#1}%
  }%
  {%
    \PackageError{glossaries}{Conflicting sort options:^^J
      \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
      \string\printnoidxglossary[sort=nocase]}{}%
  }%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
  \def\@glo@sortinglist{}%
  \forglseentries[#1]{\@gls@thislabel}%
  {%
    \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
    {%
      \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
    }%
    {%
      }%
    }%
  }%
  \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
  \ifinlistcs{#1}{\@glsref@{\@glo@type}}%
  {}%
  {%
    \listcsadd{\@glsref@{\@glo@type}}{#1}%
  }%
  \ifcsdef{\@glo@sortingchildren@#1}%
  {%
    \@glo@addchildren{\@glo@type}{#1}%
  }%
  {}%
}
\newcommand*{\@glo@sortmacro@use}[1]{%
\newcommand*{\@print@noidx@glossary}{%
  \ifcsdef{\@glsref@{\@glo@type}}%
  {%
    \ifcsdef{\@glo@sortmacro@{\@glo@sorttype}}%
    {%
      \csuse{\@glo@sortmacro@{\@glo@sorttype}}{\@glo@type}%
    }%
    {%
      \PackageError{glossaries}{Unknown sort handler ‘\@glo@sorttype’}{}%
    }%
    \glossarysection[\glossarytoctitle]{\glossarytitle}%
    \glossarypreamble
    \def\@gls@currentlettergroup{}%
  }%
}

```

```

\begin{theglossary}%
\glossaryheader
\glsresetentrylist
\forlistcsloop{\@gls@noidx@do}{\@glsref@{\@glo@type}%
\end{theglossary}%
\glossarypostamble
}%
{%
\@gls@noref@warn{\@glo@type}%
}%
}
\def\glo@grabfirst#1#2\@nil{%
\def\@gls@firsttok{#1}%
\ifdefempty\@gls@firsttok
{%
\def\@glo@thislettergrp{0}%
}%
{%
\@onelevel@sanitize\@gls@firsttok
\expandafter\@glo@grabfirst\@gls@firsttok{ }\@nil
}%
}
\def\@glo@grabfirst#1#2\@nil{%
\ifdefempty\@glo@thislettergrp
{%
\def\@glo@thislettergrp{glssymbols}%
}%
{%
\count@=\uccode' #1\relax
\ifnum\count@=0\relax
\def\@glo@thislettergrp{glssymbols}%
\else
\ifdefstring\@glo@sorttype{case}%
{%
\count@=' #1\relax
}%
{%
}%
}%
\edef\@glo@thislettergrp{\the\count@}%
\fi
}%
}
\newcommand{\@gls@noidx@do}[1]{%
\global\letcs{\@gls@loclist}{glo@glstdetoklabel{#1}@loclist}%
\ifglshasparent{#1}%
{%
\gls@level=\csuse{glo@glstdetoklabel{#1}@level}\relax
\ifdefvoid{\@gls@loclist}
{%
\subglossentry{\gls@level}{#1}{}%

```



```

}%
{%
  \subglossentry{\gls@level}{#1}%
  {%
    \glsnoidxprenumberlist{#1}%
    \glossaryentrynumbers{\glsnoidxloclist{\@gls@loclist}}%
  }%
}%
}%
{%
  \letcs{\@gls@sort}{glo@glsetoklabel{#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@sref@#1}{\csgdef{\@gls@sref@#1}{}}{}%
    \ifinlistcs{#2}{\@gls@sref@#1}%
    {}%
    {\listcsgadd{\@gls@sref@#1}{#2}}%
    \ifcsundef{glo@glstdetoklabel{#2}@loclist}%
    {\csgdef{glo@glstdetoklabel{#2}@loclist}{}}%
    {}%
    {\listcsgadd{glo@glstdetoklabel{#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{\@gls@style@#1}%
  {%
    \PackageError{glossaries}%
    {Glossary style ‘#1’ undefined}{}%
  }%
  {%
    \def\@glossarystyle{\setglossarystyle{#1}}%
  }%
}
\define@choicekey{printgloss}{numberedsection}%
[\@gls@numberedsection@val\@gls@numberedsection@nr]%

```

```

{false,nolabel,autolabel,nameref}[nolabel]%
{%
\ifcase\gls@numberedsection@nr\relax
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{}%
\or
\renewcommand*{\@@glossarysecstar}{}%
\renewcommand*{\@@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
\or
\renewcommand*{\@@glossarysecstar}{*}%
\renewcommand*{\@@glossaryseclabel}{%
\protected@edef\@currentlabelname{\glossarytoctitle}%
\label{\glsautoprefix\@glo@type}}%
\fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
\csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
\csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
\csuse{glsentrycounter#1}%
\@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
\csuse{glssubentrycounter#1}%
\@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}{gls}{nonumberlist}[true]{%
\ifglsnonumberlist
\def\glossaryentrynumbers##1{%
}
\else
\def\glossaryentrynumbers##1{##1}%
\fi}
\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
\PackageError{glossaries}{‘sort’ key not permitted with
\string\printglossary}%
{The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
\def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
\gdef\glossaryentrynumbers##1{%

```

```

        \glsresetentrylist
    }%
}
\newcommand*{\@glsnextpages}{%
    \gdef\glossaryentrynumbers##1{%
        ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
    \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
    \ifglssubentrycounter
        \setcounter{glossarysubentry}{0}%
    \fi
}
\newcommand*{\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*{\glssentryitem}[1]{%
  \ifglssentrycounter
    \glssstepentry{#1}\glssentrycounterlabel
  \else
    \glssresetsubentrycounter
  \fi
}
\newcommand*{\glsssubentryitem}[1]{%
  \ifglsssubentrycounter
    \glssstepsubentry{#1}\glsssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glstdetoklabel{#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][1]{%
  \def\@glo@counterprefix{#1}%
  \ifx\@glo@counterprefix\@empty
    \def\@glo@counterprefix{.}%
  \else
    \def\@glo@counterprefix{.#1.}%
  \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*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\@xp@glshypernumber{#1}}
}
\newrobustcmd{\@xp@glshypernumber}[1]{%
  \@glshypernumber#1\nohyperpage{}\@nil
}

```

```

}
\def\@gls\hypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\#1\%
    \else
      \gls@delimR#1\delimR\delimR\%
    \fi
    \ifx\#2\%
      \else
        #2%
      \fi
      \ifx\#3\%
        \else
          \@gls\hypernumber#3\@nil
        \fi
      \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{\gls@hypersuffix{#1}}}
\newcommand*{\hypersf}[1]{\textsf{\gls@hypersuffix{#1}}}
\newcommand*{\hypertt}[1]{\texttt{\gls@hypersuffix{#1}}}
\newcommand*{\hyperbf}[1]{\textbf{\gls@hypersuffix{#1}}}
\newcommand*{\hypermd}[1]{\textmd{\gls@hypersuffix{#1}}}
\newcommand*{\hyperit}[1]{\textit{\gls@hypersuffix{#1}}}
\newcommand*{\hypersl}[1]{\textsl{\gls@hypersuffix{#1}}}
\newcommand*{\hyperup}[1]{\textup{\gls@hypersuffix{#1}}}
\newcommand*{\hypersc}[1]{\textsc{\gls@hypersuffix{#1}}}
\newcommand*{\hyperemph}[1]{\emph{\gls@hypersuffix{#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]{#1}{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}

```

```

\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1{#3}{#4}[#5]}{#2{#3}{#4}}{}}%
}
\newcommand{\acrfullformat}[2]{#1\glsspace(#2)}
\newrobustcmd{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@Acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@Acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\@ns@acrfull}
\newcommand*\ns@acrfull[2][]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}}{}}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\Acrfull}{\@gls@hyp@opt\@ns@Acrfull}
\glsmfuaddmap{\acrfull}{\Acrfull}
\newcommand*\ns@Acrfull[2][]{%
  \new@ifnextchar[{\@Acrfull{#1}{#2}}%
    {\@Acrfull{#1}{#2}}{}}%
}
\def\@Acrfull#1#2[#3]{%
  \Acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*{\ACRfull}{\@gls@hyp@opt\@ns@ACRfull}
\newcommand*\ns@ACRfull[2][]{%
  \new@ifnextchar[{\@ACRfull{#1}{#2}}%
    {\@ACRfull{#1}{#2}}{}}%
}

```

```

\def\@ACRfull#1#2[#3]{%
  \ACRfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\acrfullpl{\@gls@hyp@opt\@ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}{%
    {\@acrfullpl{#1}{#2}}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\Acrfullpl{\@gls@hyp@opt\@ns@Acrfullpl}
\glsmfuaddmap{\acrfullpl}{\Acrfullpl}
\newcommand*\ns@Acrfullpl[2][{}]{%
  \new@ifnextchar[{\@Acrfullpl{#1}{#2}}{%
    {\@Acrfullpl{#1}{#2}}[]}%
}
\def\@Acrfullpl#1#2[#3]{%
  \Acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\ACRfullpl{\@gls@hyp@opt\@ns@ACRfullpl}
\newcommand*\ns@ACRfullpl[2][{}]{%
  \new@ifnextchar[{\@ACRfullpl{#1}{#2}}{%
    {\@ACRfullpl{#1}{#2}}[]}%
}
\def\@ACRfullpl#1#2[#3]{%
  \ACRfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acnameformat[2]{\acronymfont{#1}}
\newtoks\glskeylisttok
\newtoks\glslabeltok
\newtoks\glsshorttok
\newtoks\glslongtok
\newcommand*\newacronymhook{}
\newcommand*\SetGenericNewAcronym{%
  \let\@Gls@entryname\@Gls@acentryname
  \renewcommand{\newacronym}[4][{}]{%
    \ifdefempty{\@glsacronymlists}%
    {%
      \def\@glo@type{\acronymtype}%
      \setkeys{glossentry}{##1}%
      \DeclareAcronymList{\@glo@type}%
    }%
    {}%
    \glskeylisttok{##1}%
    \glslabeltok{##2}%
    \glsshorttok{##3}%
    \glslongtok{##4}%
  }%
}

```

```

\newacronymhook
\protected@edef\@do@newglossaryentry{%
  \noexpand\newglossaryentry{\the\glslabeltok}%
  {%
    type=\acronymtype,%
    name={\expandonce{\acronymentry{##2}}},%
    sort={\acronymsort{\the\glsshorttok}{\the\glslongtok}},%
    text={\the\glsshorttok},%
    short={\the\glsshorttok},%
    shortplural={\the\glsshorttok\noexpand\acrpluralsuffix},%
    long={\the\glslongtok},%
    longplural={\the\glslongtok\noexpand\glspluralsuffix},%
    \GenericAcronymFields,%
    \the\glskeylisttok
  }%
}%
\@do@newglossaryentry
}%
\renewcommand*{\acrfullfmt}[3]{%
  \glslink[##1]{##2}{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\Genacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glssupercase{\genacrfullformat{##2}{##3}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glssupercase{\genplacrfullformat{##2}{##3}}}%
\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{%
        \defglentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
}%
}
\newcommand*{\newacronymstyle}[3]{%
    \ifcsdef{@glsacr@dispstyle@#1}%
    {%
        \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
    }%
    {%
        \csdef{@glsacr@dispstyle@#1}{#2}%
        \csdef{@glsacr@styledefs@#1}{#3}%
    }%
}
\newcommand*{\renewacronymstyle}[3]{%
    \ifcsdef{@glsacr@dispstyle@#1}%
    {%
        \csdef{@glsacr@dispstyle@#1}{#2}%
        \csdef{@glsacr@styledefs@#1}{#3}%
    }%
    {%
        \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
    }%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
    \renewcommand*{\genacrfullformat}[2]{%
        \glstrylong{##1}##2\space
        (\protect\firstacronymfont{\glstryshort{##1}})%
    }%
    \renewcommand*{\Genacrfullformat}[2]{%
        \glstrylong{##1}##2\space
        (\protect\firstacronymfont{\glstryshort{##1}})%
    }%
    \renewcommand*{\genplacrfullformat}[2]{%
        \glstrylongpl{##1}##2\space
        (\protect\firstacronymfont{\glstryshortpl{##1}})%
    }%
    \renewcommand*{\Genplacrfullformat}[2]{%
        \glstrylongpl{##1}##2\space

```

```

(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glsentrylong{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glsentrylongpl{##1}##2\glsacspace{##1}%
(\protect\firstacronymfont{\glentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglshaslong{\glslabel}{\glsacrfmt}{\glsentryfmt}%
}%
{%
\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
  (\glsentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
  \protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
  \protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
  (\glsentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
  \GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-short}%
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%

```

```

\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDispStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\glspacespace{##1}(\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
\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}\glinsert
}%
{%
\Glentrylongpl{\glslabel}\glinsert
}%
{%
\glsupercase
{\glentrylongpl{\glslabel}\glinsert}%
}%
}%
}

```

```

}%
{%
  \glscapscase
  {%
    \glentrylong{\glslabel}\glsinsert
  }%
  {%
    \Glsentrylong{\glslabel}\glsinsert
  }%
  {%
    \glssupercase
    {\glentrylong{\glslabel}\glsinsert}%
  }%
}%
}%
{%
  \glsgenentryfmt
}%
}%
{\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glentrylong{##2}##3\space
      (\acronymfont{\glentryshort{##2}})}}%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
      (\acronymfont{\glentryshort{##2}})}}%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glssupercase{\glentrylong{##2}##3\space
        (\acronymfont{\glentryshort{##2}})}}}%
  \renewcommand*{\acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\glentrylongpl{##2}##3\space
      (\acronymfont{\glentryshortpl{##2}})}}%
  \renewcommand*{\Acrfullplfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylongpl{##2}##3\space
      (\acronymfont{\glentryshortpl{##2}})}}%
  \renewcommand*{\ACRfullplfmt}[3]{%
    \glslink[##1]{##2}{%
      \glssupercase{\glentrylongpl{##2}##3\space
        (\acronymfont{\glentryshortpl{##2}})}}}%
  \renewcommand*{\glsentryfull}[1]{%
    \glentrylong{##1}\space(\acronymfont{\glentryshort{##1}})%
  }%
  \renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glentryshort{##1}})%
  }%
  \renewcommand*{\glsentryfullpl}[1]{%

```

```

\glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
\Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
\GlsUseAcrEntryDispStyle{dua}%
}%
{%
\GlsUseAcrStyleDefs{dua}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
\renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\glshyperfirstfalse
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glsentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glsentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink{##1}{##2}{\acronymfont{\glsentryshort{##2}}##3\space
(\glsentrylong{##2})}%
}

```

```

\renewcommand*{\Acrfullfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
    (\Glsentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsupercase{\acronymfont{\Glsentryshort{##2}}##3\space
      (\Glsentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
    (\Glsentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
    (\Glsentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glsupercase{\acronymfont{\Glsentryshortpl{##2}}##3\space
      (\Glsentrylongpl{##2})}}}%
\renewcommand*{\glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\Glsentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
  \acronymfont{\Glsentryshort{##1}}\space(\Glsentrylong{##1})}%
\renewcommand*{\glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\Glsentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
  \acronymfont{\Glsentryshortpl{##1}}\space(\Glsentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\Glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsc{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand{\acronymentry}[1]{\acronymfont{\Glsentryshort{##1}}}
  \renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
  \renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{footnote}%
}%

```

```

{%
  \GlsUseAcrStyleDefs{footnote}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
  \newacronymstyle{footnote-sc-desc}%
  {%
    \GlsUseAcrEntryDisplayStyle{footnote-sc}%
  }%
  {%
    \GlsUseAcrStyleDefs{footnote-sc}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymsort}[2]{##2}%
    \renewcommand*{\acronymentry}[1]{%
      \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
    }
    \newacronymstyle{footnote-sm-desc}%
    {%
      \GlsUseAcrEntryDisplayStyle{footnote-sm}%
    }%
    {%
      \GlsUseAcrStyleDefs{footnote-sm}%
      \renewcommand*{\GenericAcronymFields}{}%
      \renewcommand*{\acronymsort}[2]{##2}%
      \renewcommand*{\acronymentry}[1]{%
        \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
      }
      \newcommand*{\DefineAcronymSynonyms}{%
        \let\acs\acrshort
        \let\Acs\Acrshort
        \let\acsp\acrshortpl
        \let\Acsp\Acrshortpl
        \let\acl\acrlong
        \let\aclp\acrlongpl
        \let\Acl\Acrlong
        \let\Aclp\Acrlongpl
        \let\acf\acrfull
        \let\acfp\acrfullpl
        \let\Acf\Acrfull
        \let\Acfp\Acrfullpl
        \let\ac\gls
        \let\Ac\Gls
        \let\acp\glsp
        \let\Acp\Glsp
      }
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      \fi
    }
  }

```

```

\newcommand*\SetDefaultAcronymDisplayStyle}[1]{%
  \def\glsentryfmt[#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}
\@gls@loadlist
\@gls@loadlong
\@gls@loadsuper
\@gls@loadtree
\ifx\@glossary@default@style\relax
\else
  \setglossarystyle{\@glossary@default@style}
\fi
\newcommand*\showgloparent}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@parent\endcsname
}
\newcommand*\showglolevel}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@level\endcsname
}
\newcommand*\showglotext}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@text\endcsname
}
\newcommand*\showgloplural}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@plural\endcsname
}
\newcommand*\showglofirst}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@first\endcsname
}

```

```

}
\newcommand*{\showglofirstpl}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@firstpl\endcsname
}
\newcommand*{\showgloftype}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@type\endcsname
}
\newcommand*{\showglocounter}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@counter\endcsname
}
\newcommand*{\showglouserii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userii\endcsname
}
\newcommand*{\showglouseriii}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriii\endcsname
}
\newcommand*{\showglouseriv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@useriv\endcsname
}
\newcommand*{\showglouserv}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@userv\endcsname
}
\newcommand*{\showglouservi}[1]{%
  \expandafter\show\csname glo@\glsdetoklabel{#1}@uservi\endcsname
}
\newcommand*{\showglongname}[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*{\showglooclist}[1]{%
\expandafter\show\csname glo@\glsdetoklabel{#1}@oclist\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
}
}

```

9.2 Rollback v4.58 (glossaries-2025-03-19.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}
\DeclareRelease{v4.52}{2022-11-03}{glossaries-2022-11-03.sty}
\DeclareRelease{v4.54}{2024-04-03}{glossaries-2024-04-03.sty}
\DeclareCurrentRelease{v4.58}{2025-03-19}
\ProvidesPackage{glossaries}[2025/03/19 v4.58 (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{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]{%
  \glsshowtargetsymbols\marginpar{\glsshowtargetsymbols\glsshowtargetfont #1}}
\newcommand*{\glsshowtargetsymbols}{\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]{%
  \ifstrepty{#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]{%
\glssavenumberlistfalse
\newcommand*\@glo@seeautonumberlist{%
\@gls@declareoption{seeautonumberlist}{%
  \renewcommand*{\@glo@seeautonumberlist}{%
    \def\@glo@prefix{\glsnextpages}%
  }%
}
}
\define@boolkey{glossaries.sty}[gls]{esclocations}[true]{%
\glseclocationsfalse
\newcommand*\@gls@loadlong{\RequirePackage{glossary-long}}
\@gls@declareoption{nolong}{\renewcommand*{\@gls@loadlong}{}}
\IfFileExists{supertabular.sty}{%
  \newcommand*\@gls@loadsuper{\RequirePackage{glossary-super}}}%
  \newcommand*\@gls@loadsuper{}%
\@ifpackageloaded{xtab}{\renewcommand*\@gls@loadsuper{}}{}
\@gls@declareoption{nosuper}{\renewcommand*\@gls@loadsuper{}}
\newcommand*\@gls@loadlist{\RequirePackage{glossary-list}}
\@gls@declareoption{nolist}{%
  \renewcommand*\@gls@loadlist{%
    \ifdefstring{\@glossary@default@style}{list}%
    {\let\@glossary@default@style\relax}%
  }%
}%
}
\newcommand*\@gls@loadtree{\RequirePackage{glossary-tree}[=v4.58]}
\@gls@declareoption{notree}{\renewcommand*\@gls@loadtree{}}
\@gls@declareoption{nostyles}{%
  \renewcommand*\@gls@loadlong{}%
  \renewcommand*\@gls@loadsuper{}%
  \renewcommand*\@gls@loadlist{}%
  \renewcommand*\@gls@loadtree{}%
  \let\@glossary@default@style\relax
}
\newcommand*\@gls@postdescription{%
  \ifglsnopostdot\else.\spacefactor\sfcode'\. \fi
}
\define@boolkey{glossaries.sty}[gls]{nopostdot}[true]{%
\glsnopostdotfalse
\define@boolkey{glossaries.sty}[gls]{nogroupskip}[true]{%
\glsnogroupskipfalse
\define@boolkey{glossaries.sty}[gls]{ucmark}[true]{%
\@ifclassloaded{memoir}
{%
  \glsucmarktrue
}%
}%
}

```

```

\glsucmarkfalse
}
\newcommand*{\@gls@define@glossaryentrycounter}{%
\ifglentrycounter
\ifundef\c@glossaryentry
{%
\ifx\@gls@counterwithin\@empty
\newcounter{glossaryentry}%
\else
\newcounter{glossaryentry}[\@gls@counterwithin]%
\fi
\def\theHglossaryentry{\currentglossary.\theglossaryentry}%
}%
}%
\fi
}
\define@boolkey{glossaries.sty}[gls]{entrycounter}[true]{%
\glentrycounterfalse
\define@key{glossaries.sty}{counterwithin}{%
\renewcommand*{\@gls@counterwithin}{#1}%
\glentrycountertrue
\@gls@define@glossaryentrycounter
}
\newcommand*{\@gls@counterwithin}{}
\newcommand*{\@gls@define@glossarysubentrycounter}{%
\ifundef\c@glossarysubentry
{%
\ifglssubentrycounter
\ifglentrycounter
\newcounter{glossarysubentry}[glossaryentry]%
\else
\newcounter{glossarysubentry}%
\fi
\def\theHglossarysubentry{\currentglossary.\currentglssubentry.\theglossarysubentry}%
\fi
}%
}%
}
\define@boolkey{glossaries.sty}[gls]{subentrycounter}[true]{%
\glssubentrycounterfalse
\newcommand{\glsindexingsetting}{\ifglxindy 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]{%

```

```

\ExplSyntaxOn
\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}%
      \tl_gset:cV { glo@##2@sort } \@glo@sort
    }%
    \def\@gls@setsort##1{%
      \let\@glo@check@sortallowed\@gobble
    }
  \@gls@setupsort@standard
\newcommand*\glssortnumberfmt[1]{%
  \int_compare:nNnT { #1 } < { 100000 } { 0 }
  \int_compare:nNnT { #1 } < { 10000 } { 0 }
  \int_compare:nNnT { #1 } < { 1000 } { 0 }
  \int_compare:nNnT { #1 } < { 100 } { 0 }
  \int_compare:nNnT { #1 } < { 10 } { 0 }
  \int_eval:n { #1 }
}
\newcommand*{\@gls@setupsort@def}{%
  \def\do@glo@storeentry{\@glo@storeentry}%
  \def\@gls@defsortcount##1{%
    \int_new:c { g__glossary_ ##1 _sortcount_int }
  }%
  \def\@gls@defsort##1##2{%
    \int_if_exist:cF { g__glossary_ ##1 _sortcount_int }
    {
      \@gls@defsortcount { ##1 }
    }
    \int_gincr:c { g__glossary_ ##1 _sortcount_int }
    \tl_gset:ce { glo@##2@sort }
    {
      \exp_args:Nc \glssortnumberfmt
        { g__glossary_ ##1 _sortcount_int }
    }
  }%
  \def\@gls@setsort##1{%
    \let\@glo@check@sortallowed\@gobble
  }
\newcommand*{\@gls@setupsort@use}{%
  \let\do@glo@storeentry\@gobble
  \def\@gls@defsortcount##1{%
    \int_new:c { g__glossary_ ##1 _sortcount_int }
  }%
  \def\@gls@defsort##1##2{%

```

```

\tl_gset:cn { glo@ ##2 @sort } { }
}%
\def\@gls@setsort##1{%
\tl_if_empty:cF { glo@ ##1 @parent }
{
\exp_args:nv \@gls@setsort { glo@ ##1 @parent }
}
\tl_if_empty:cT { glo@ ##1 @sort }
{
\int_gincr:c
{
g__glossary_
\tl_use:c { glo@ ##1 @type }
_sortcount_int
}
\tl_gset:ce { glo@ ##1 @sort }
{
\exp_args:Nc \glssortnumberfmt
{
g__glossary_
\tl_use:c { glo@ ##1 @type }
_sortcount_int
}
}
\@glo@storeentry{##1}%
}
}%
\let\@glo@check@sortallowed\@gobble
}
\ExplSyntaxOff
\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]{%
  \ifglssacronym
    \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{%
  \ifglssacronym
    \providecommand*\printacronyms[1][1]{%
      \printglossary[type=\acronymtype,#1]%
    }%
  \fi
}
\newcommand*{\@gls@do@acronymsdef}{}
\@gls@declareoption{acronyms}{%
  \glssacronymtrue
  \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*{\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}{\@gls@noexpandfield{name}}
\newcommand*{\@gls@sanitizesymbol}{\@gls@noexpandfield{symbol}}
\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}{\@gls@noexpandfield{symbol}}
\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
}
\ExplSyntaxOn
\newcommand*{\@gls@noidx@nosanitizesort}
{
  \cs_if_exist:NTF \datatool_sort_preprocess:Nn
  {
    \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
        }
    }
}
{
\@@gls@old@noidx@nosanitizesort
}
}
\ExplSyntaxOff
\newcommand*{\@@gls@old@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{%
\bgrou
\glsnoidxstripaccents
\protected@xdef\@@glo@sort{#1#2}%
\egrou
\let\@glo@sort\@@glo@sort
}
\newcommand*\glsnoidxstripaccents{%
\let\IeC\@firstofone
\let\add@accent@\@secondoftwo
\let\@text@composite@x\@secondoftwo
\let\@tabacckludge\@secondoftwo
\expandafter\def\csname \encodingdefault-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname OT1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname T1-cmd\endcsname##1##2##3{##3}%
\expandafter\def\csname PD1-cmd\endcsname##1##2##3{##3}%
\let\'@\@firstofone
\let\'@\@firstofone
\let\~@\@firstofone
\let\"@\@firstofone
\let\u@\@firstofone
\let\t@\@firstofone
\let\d@\@firstofone
\let\r@\@firstofone
\let\=@\@firstofone
\let\.\@firstofone
\let\~@\@firstofone
\let\v@\@firstofone
\let\H@\@firstofone
\let\c@\@firstofone

```

```

\let\b\@firstofone
\let\a\@secondoftwo
\def\AE{AE}%
\def\ae{ae}%
\def\OE{OE}%
\def\oe{oe}%
\def\AA{AA}%
\def\aa{aa}%
\def\L{L}%
\def\l{l}%
\def\O{O}%
\def\o{o}%
\def\SS{SS}%
\def\ss{ss}%
\def\th{th}%
\def\TH{TH}%
\def\dh{dh}%
\def\DH{DH}%
}
\@ifl@t@r{fmtversion{2019/10/01}}
{%
\appto\glsnoidxstripaccents{\let\UTFviii@two@octets\UTFviii@two@octets@combine}%
}
{}
\newcommand{\@gls@removedoption}[1]{%
\PackageError{glossaries}{obsolete package option ‘#1’ has been
removed. Rollback required or use a newer alternative}
{Try \string\usepackage[#1]{glossaries}[=v4.46]}%
}
\define@boolkey[gls]{sanitize}{description}[true]{%
\GlossariesWarning{sanitize={description} package option deprecated}%
\ifgls@sanitize@description
\glssetnoexpandfield{desc}%
\glssetnoexpandfield{descplural}%
\else
\glssetexpandfield{desc}%
\glssetexpandfield{descplural}%
\fi
}
\define@boolkey[gls]{sanitize}{name}[true]{%
\GlossariesWarning{sanitize={name} package option deprecated}%
\ifgls@sanitize@name
\glssetnoexpandfield{name}%
\else
\glssetexpandfield{name}%
\fi
}
\define@boolkey[gls]{sanitize}{symbol}[true]{%
\GlossariesWarning{sanitize={symbol} package option deprecated}%
\ifgls@sanitize@symbol

```

```

        \glsssetnoexpandfield{symbol}%
        \glsssetnoexpandfield{symbolplural}%
    \else
        \glsssetexpandfield{symbol}%
        \glsssetexpandfield{symbolplural}%
    \fi
}
\define@boolkey{glossaries.sty}[gls]{sanitizesort}[true]{%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesorttrue
            \glsssetnoexpandfield{sortvalue}%
        }%
    \else
        \glsssetexpandfield{sortvalue}%
        \renewcommand*{\@gls@noidx@setsanitizesort}{%
            \glsssanitizesortfalse
            \glsssetexpandfield{sortvalue}%
        }%
    \fi
}
\glsssanitizesorttrue
\glsssetnoexpandfield{sortvalue}%
\newcommand*{\@gls@noidx@setsanitizesort}{%
    \glsssanitizesortfalse
    \glsssetexpandfield{sortvalue}%
}
\define@choicekey[gls]{sanitize}{sort}[true,false][true]{%
    \setbool{glsssanitizesort}{#1}%
    \ifglsssanitizesort
        \glsssetnoexpandfield{sortvalue}%
    \else
        \glsssetexpandfield{sortvalue}%
    \fi
    \GlossariesWarning{sanitize={sort} package option
        deprecated. Use sanitizesort instead}%
}
\define@key{glossaries.sty}{sanitize}[description=true,symbol=true,name=true]{%
    \ifthenelse{\equal{#1}{none}}{%
        {%
            \GlossariesWarning{sanitize package option deprecated}%
            \glsssetexpandfield{name}%
            \glsssetexpandfield{symbol}%
            \glsssetexpandfield{symbolplural}%
            \glsssetexpandfield{desc}%
            \glsssetexpandfield{descplural}%
        }%
    }{%
        \setkeys[gls]{sanitize}{#1}%
    }
}

```

```

}%
}
\newif\ifglstranslate
\newcommand*\@gls@usetranslator{%
  \@ifpackageloaded{polyglossia}%
  {%
    \let\glsifusetranslator\@secondoftwo
  }%
  {%
    \@ifpackageloaded{babel}%
    {%
      \IfFileExists{translator.sty}%
      {%
        \RequirePackage{translator}%
        \let\glsifusetranslator\@firstoftwo
      }%
    }%
  }%
}%
}
\newcommand{\glsifusedtranslatordict}[3]{%
  \glsifusetranslator
  {\ifcsdef{ver@glossaries-dictionary-#1.dict}{#2}{#3}}%
  {#3}%
}
\@gls@declareoption{nottranslate}{%
  \glstranslatefalse
  \let\@gls@usetranslator\relax
  \let\glsifusetranslator\@secondoftwo
}
\define@choicekey{glossaries.sty}{translate}%
[\gls@translate@val\gls@translate@nr]%
{true,false,babel}[true]%
{%
  \ifcase\gls@translate@nr\relax
  \glstranslatetrue
  \renewcommand*\@gls@usetranslator{%
    \@ifpackageloaded{polyglossia}%
    {%
      \let\glsifusetranslator\@secondoftwo
    }%
  }%
  \@ifpackageloaded{babel}%
  {%
    \IfFileExists{translator.sty}%
    {%
      \RequirePackage{translator}%
      \let\glsifusetranslator\@firstoftwo
    }%
  }%
}

```



```

        {}%
    }%
    {}%
} %
} %
\or
\glstranslatefalse
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\or
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
\fi
}
\glstranslatefalse
\let\glsifusetranslator\@secondoftwo
\@ifpackageloaded{translator}%
{%
\glstranslatetrue
\let\glsifusetranslator\@firstoftwo
}%
{%
\@for\gls@thissty:=tracklang,babel,ngerman,polyglossia\do
{
\@ifpackageloaded{\gls@thissty}%
{%
\glstranslatetrue
\@endfortrue
}%
}%
}
}
\newcommand{\@gls@requested@languages}{}
\define@key{glossaries.sty}{languages}{%
\renewcommand{\@gls@requested@languages}{#1}%
\glstranslatetrue
\let\@gls@usetranslator\relax
\let\glsifusetranslator\@secondoftwo
}
\define@key{glossaries.sty}{locales}{%
\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]{%
\glsxindy@glslnumberstrue
\def\@xdy@main@language{\@language}%
\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@filesw #1\fi}
\@glS@declareoption{disablemakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘disablemakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘disablemakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
    {%
      \renewcommand{\@domakeglossaries}[1]{%
        \PackageInfo{glossaries}{\string\makeglossaries\space and
          \string\makenoidxglossaries\space have been disabled}%
      }%
    }%
  }%
}
\@glS@declareoption{restoremakegloss}{%
  \ifdefequal\makeglossaries\@no@makeglossaries
  {%
    \GlossariesWarning{Option ‘restoremakegloss’ has no effect
      (\string\makenoidxglossaries\space has already been used)}%
  }%
  {%
    \ifdefequal\@makeglossary\@gobble
    {%
      \GlossariesWarning{Option ‘restoremakegloss’ has no effect
        (\string\makeglossaries\space has already been used)}%
    }%
    {%
      \PackageInfo{glossaries}{\string\makeglossaries\space and
        \string\makenoidxglossaries\space have been restored}%
      \let\@domakeglossaries\@firstofone
    }%
  }%
}

```

```

}%
}
\newcommand*{\do@write@glslabels}{%
\AtEndDocument{\do@write@glslabels}%
\let\do@write@glslabels\relax
}
\newcommand*{\do@write@glslabels}{%
\newwrite\@gls@labelsfile
\immediate\openout\@gls@labelsfile=\jobname.glslabels
\protected@edef\@gls@fulllist{\@glo@types}%
\ifdefempty{\@ignored@glossaries}{}%
{\protected@eappto\@gls@fulllist{\@ignored@glossaries}}%
\forallglsentries[\@gls@fulllist]{\@glsentry}%
{%
\ifdefempty{\@glsentry}{}%
{%
\ifx\do@write@glslabelsname\@gobble
\def\@gls@thisentryname{}%
\else
\letcs{\@gls@thisentryname}{glo\@glsdetoklabel{\@glsentry}@name}%
\@onelevel@sanitize\@gls@thisentryname
\fi
\immediate\write\@gls@labelsfile
{\@glsentry\do@write@glslabelsname{\@gls@thisentryname}}%
}%
}%
\immediate\closeout\@gls@labelsfile
}
\newcommand*{\do@write@glslabelsname}[1]{%
\begingroup
\catcode'\^^I12
\gdef\@write@glslabelsname#1{\^^I#1}
\endgroup
\@gls@declareoption{writeglslabels}{\do@write@glslabels
\let\do@write@glslabelsname\@gobble}
\@gls@declareoption{writeglslabelnames}{\do@write@glslabels
\let\do@write@glslabelsname\@write@glslabelsname}
\newif\ifglsautomake
\newcommand{\gls@automake@nr}{1}
\newcommand{\gls@automake@makegloss}{%
\newcommand{\gls@automake@option}{automake=\gls@automake@val}
\newcommand*{\@gls@do@automake@err}{%
\PackageError{glossaries}{You must use
\string\makeglossaries\space with \gls@automake@option}
{%
Either remove the \gls@automake@option\space setting or
add \string\makeglossaries\space to your document preamble.%
}%
}%
}%
\define@choicekey{glossaries.sty}{automake}%

```

```

[\gls@automake@val\gls@automake@nr]{true,false,immediate,makegloss,lite,delayed}[immediate]{%
\ifcase\gls@automake@nr\relax
  \GlossariesWarning{automake=true deprecated. Use automake=delayed for a delayed write}%
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{}%
\or
  \glsautomakefalse
  \renewcommand{\gls@automake@makegloss}{}%
\or
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{}%
\or
  \glsautomaketrue
  \renewcommand{\gls@automake@makegloss}{makeglossaries \@gls@run@output@dir{-d}}%
\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 \@gls@run@output@dir{-d}}%
  \let\@gls@doautomake\@gls@do@automake@err
}
\@gls@declareoption{automakeglosslite}{%
  \glsautomaketrue
  \renewcommand{\gls@automake@option}{automakeglosslite}%
  \def\gls@automake@val{lite}%
  \def\gls@automake@nr{4}%
  \renewcommand{\gls@automake@makegloss}{makeglossaries-lite}%
  \let\@gls@doautomake\@gls@do@automake@err
}
\define@boolkey{glossaries.sty}[gls]{savewrites}[true]{%
  \ifglssavewrites

```

```

        \renewcommand*{\glswritefiles}{\@glswritefiles}%
    \else
        \let\glswritefiles\@empty
    \fi
}
\glssavewritesfalse
\let\glswritefiles\@empty
\define@boolkey{glossaries.sty}[gls]{compatible-3.07}[true]{%
    \ifbool{glscompatible-3.07}%
    {%
        \@gls@removedoption{compatible-3.07}%
    }%
    {}%
}
\boolfalse{glscompatible-3.07}
\define@boolkey{glossaries.sty}[gls]{compatible-2.07}[true]{%
    \ifbool{glscompatible-2.07}%
    {%
        \@gls@removedoption{compatible-2.07}%
    }%
    {}%
}
\boolfalse{glscompatible-2.07}
\let\gls@original@makeglossary\makeglossary
\let\gls@original@glossary\glossary
\newcommand*{\@gls@override@glossary}[1][main]{%
    \GlossariesWarning{Use of \string\glossary\space with
glossaries.sty is deprecated. \MessageBreak Indexing should be performed
with the user level \MessageBreak commands, such as \string\gls\space or
\string\glsadd. If you need the \MessageBreak original definition of
\string\glossary\space use the package \MessageBreak options
kernelglossredefs=false (to restore the \MessageBreak former definition of
\string\glossary) and nomain (if the \MessageBreak file extensions cause a
conflict)}}%
    \gls@glossary{#1}%
}
\define@choicekey{glossaries.sty}{kernelglossredefs}%
[\gls@kernelglossredefs@val\gls@kernelglossredefs@nr]{true,false,nowarn}[true]%
{%
    \ifcase\gls@kernelglossredefs@nr\relax
        \def\glossary{\@gls@override@glossary}%
        \def\makeglossary{%
            \GlossariesWarning{Use of \string\makeglossary\space with
glossaries.sty is deprecated. Use \string\makeglossaries\space
instead. If you need the original definition of
\string\makeglossary\space use the package options
kernelglossredefs=false (to prevent redefinition of
\string\makeglossary) and nomain (if the file extensions cause a
conflict)}}%
            \makeglossaries

```

```

}%
\or
\let\glossary\gls@original@glossary
\let\makeglossary\gls@original@makeglossary
\or
\def\makeglossary{\makeglossaries}%
\renewcommand*{\@gls@override@glossary}[1][main]{%
\gls@glossary{##1}%
}%
\fi
}
\@gls@declareoption{symbols}{%
\let\@gls@do@symbolsdef\@gls@symbolsdef
}
\newcommand*{\@gls@do@symbolsdef}{%
\newcommand*{\@gls@symbolsdef}{%
\newglossary[slg]{symbols}{sls}{slo}{\glssymbolsgroupname}%
\newcommand*{\printsymbols}[1][\printglossary[type=symbols,##1]]{%
\newcommand*{\gls@tr@set@symbols@toctitle}{%
\translatelet{\glossarytoctitle}{Symbols (glossaries)}}%
}%
}%
\@gls@declareoption{numbers}{%
\let\@gls@do@numbersdef\@gls@numbersdef
}
\newcommand*{\@gls@do@numbersdef}{%
\newcommand*{\@gls@numbersdef}{%
\newglossary[nlg]{numbers}{nls}{nlo}{\glsnumbersgroupname}%
\newcommand*{\printnumbers}[1][\printglossary[type=numbers,##1]]{%
\newcommand*{\gls@tr@set@numbers@toctitle}{%
\translatelet{\glossarytoctitle}{Numbers (glossaries)}}%
}%
}%
\@gls@declareoption{index}{%
\ifx\@gls@do@indexdef\@empty
\let\@gls@do@indexdef\@gls@indexdef
\fi
}
\@gls@declareoption{noglossaryindex}{%
\let\@gls@do@indexdef\relax
}
\newcommand*{\@gls@do@indexdef}{%
\newcommand*{\@gls@indexdef}{%
\newglossary[ilg]{index}{ind}{idx}{\indexname}%
\newcommand*{\printindex}[1][\printglossary[type=index,##1]]{%
\newcommand*{\newterm}[2][\newglossaryentry{##2}%
{type={index},name={##2},description={\nopostdesc},##1}}
\let\@gls@do@indexdef\relax
}%
}

```

```

\define@choicekey{glossaries.sty}{mfirstuc}%
[\gls@mfirstuc@val\gls@mfirstuc@nr]%
{expanded,unexpanded}%
{%
  \ifcase\gls@mfirstuc@nr
  % expanded
  \renewcommand{\glsmakefirstuc}[1]{\MFUsentencecase{#1}}%
  \or
  % unexpanded
  \renewcommand{\glsmakefirstuc}[1]{\MFUsentencecase{\unexpanded{#1}}}%
  \fi
}
\@for\CurrentOption :=\@declaredoptions\do{%
  \ifx\CurrentOption\@empty
  \else
  \@expandtwoargs
  \in@ {,\CurrentOption ,}{,\@classoptionslist,\@curroptions,}%
  \ifin@
  \@use@ption
  \expandafter \let\csname ds@\CurrentOption\endcsname\@empty
  \fi
\fi
}
\ProcessOptionsX
\newcommand*{@gls@runshell}[1]{%
  \PackageWarning{glossaries}{Ignoring attempt to run command ‘#1’}%
}
\newcommand*{@gls@run@unrestricted@shell}[1]{\@gls@runshell{#1}}
\newcommand*{@gls@run@output@dir}[1]{%
}
\newcommand*{@gls@run@output@file}[1]{#1}
\ifglsautomake
\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
    {
      \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)
      }
      \@gls@runshell { #1 }
    }
  }
}

```



```

}
\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 }
  }
  \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 }
{
  \cs_set:Npn \@gls@runshell #1 { \exp_args:Ne \write18 { #1 } }
}
{
  \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
\disable@keys{glossaries.sty}{compatible-2.07,%
xindy,xindygloss,xindynoglsnumbers,makeindex,%
acronym,translate,notranslate,languages,locales,nolong,nosuper,notree,nostyles,%
nomain,noglossaryindex}
\newcommand*{\setupglossaries}[1]{%
  \renewcommand*{\@gls@setacrstyle}{}%
  \ifglsacrshortcuts
    \def\@gls@setupshortcuts{\glsacrshortcutstrue}%
  \else
    \def\@gls@setupshortcuts{%
      \ifglsacrshortcuts
        \DefineAcronymSynonyms
      }%
  \fi
  \glsacrshortcutsfalse
  \let\@gls@do@numbersdef\relax
  \let\@gls@do@symbolssdef\relax
  \let\@gls@do@indexdef\relax
  \let\@gls@do@acronymsdef\relax
  \ifglseentrycounter
    \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}{\glsupacrpluralsuffix}
\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’}}%
}%
}%
\@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
\RequirePackage{datatool-base}
\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}
\ifglxsindy
  \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{\glslocationcstoencap}[3]{\csuse{#2}{\csuse{#1}{#3}}}
\newcommand{\delimN}{, }
\newcommand{\delimR}{--}
\newcommand*{\glossarypreamble}{%
  \csuse{@glossarypreamble@currentglossary}}%
}
\newcommand{\setglossarypreamble}[2][\glsdefaultttype]{%
  \ifglossaryexists*{#1}%
  {\csgdef{@glossarypreamble@#1}{#2}}%
  {\GlossariesWarning{Glossary ‘#1’ is not defined}}%
}
\newcommand*{\glossarypostamble}{}
\newcommand*{\glossarysection}[2][\@gls@title]{%
  \def\@gls@title{#2}%
  \ifcsundef{phantomsection}%
  {%
    \@glossarysection{#1}{#2}%
  }%
  {%
    \@p@glossarysection{#1}{#2}%
  }%
  \gls glossarymark{\glossarytoctitle}%
}
\ifcsundef{glossarymark}%
{%
  \newcommand{\gls glossarymark}[1]{\glossarymark{#1}}
}

```

```

}%
{%
  \@ifclassloaded{memoir}
  {%
    \newcommand{\gls glossarymark}[1]{%
      \ifglsucmark
        \markboth{\memUHead{#1}}{\memUHead{#1}}%
      \else
        \markboth{#1}{#1}%
      \fi
    }
  }%
  {%
    \newcommand{\gls glossarymark}[1]{%
      \ifglsucmark
        \@mkboth{\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*{@p@glossarysection}[2]{%
  \gls clearpage
  \phantomsection
  \ifdefempty\@glossarysecstar
  {%
    \csname\@glossarysec\endcsname{#2}%
  }%
  {%

```

```

\@gls@toc{#1}{\@@glossarysec}%
\csname\@@glossarysec\endcsname*{#2}%
}%
\@@glossaryseclabel
}
\newcommand*{\gls@docclearpage}{%
\ifthenelse{\equal{\@@glossarysec}{chapter}}{%
{%
\ifcsundef{cleardoublepage}%
{%
\clearpage
}%
{%
\ifcsdef{if@openright}%
{%
\if@openright
\cleardoublepage
\else
\clearpage
\fi
}%
}%
\cleardoublepage
}%
}%
}%
}%
}
\newcommand*{\gls@clearpage}{\gls@docclearpage}
\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*{\glsnomakeindexwarning}[1]{%
\GlossariesWarning{Not in makeindex mode --- ignoring \string#1}%
}
\ifglsxindy
\edef\@xdyattributes{\string"default\string"%
\fi
\ifglsxindy
\edef\@xdyattributelist{}%
\fi

```



```

\ifglxindy
  \def\@xdylocref{}
\fi
\newcommand*\@gls@ifinlist}[4]{%
  \def\@do@ifinlist##1,##2\end@doifinlist{%
    \def\@gls@listsuffix{##2}%
    \ifx\@gls@listsuffix\@empty
      #4%
    \else
      #3%
    \fi
  }%
  \@do@ifinlist,##2,##1,\end@doifinlist
}
\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]{%
    \glsnoxywarning\GlsAddXdyAttribute
  }
\fi
\newcommand*\@disabled@glssaddxdycounters{%
  \PackageError{glossaries}{\string\GlsAddXdyCounters\space
    can't be used after \string\GlsAddXdyAttribute}{Move all
    occurrences of \string\GlsAddXdyCounters\space before the first
    instance of \string\GlsAddXdyAttribute}%
}
\ifglxindy
  \newcommand*\@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{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\@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
\ifglxsindy
\newcommand\@xdycrossrefhook{}
\fi
\ifglxsindy
\newcommand*\@GlsAddXdyLocation}[3][{}]{%
\def\@gls@tmp{#1}%
\ifx\@gls@tmp\@empty
\edef\@xdyuserlocationdefs{%
\@xdyuserlocationdefs ^^J%
(define-location-class \string"#2\string"^^J\space\space
\space(:sep \string"{}\@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]{%
    \glsnxindywarning\GlsSetXdyLocationClassOrder}
\fi
\ifglxindy
  \def\@xdysortrules{}
\fi
\ifglxindy
  \newcommand*\GlsAddSortRule[2]{%
    \expandafter\toks@\expandafter{\@xdysortrules}%
    \protected@edef\@xdysortrules{\the\toks@ ^^J
      (sort-rule \string"#1\string" \string"#2\string"))}%
  }
\else
  \newcommand*\GlsAddSortRule[2]{%
    \glsnxindywarning\GlsAddSortRule}
\fi
\ifglxindy
  \def\@xdyrequiredstyles{tex}
\fi
\ifglxindy
  \newcommand*\GlsAddXdyStyle[1]{%
    \edef\@xdyrequiredstyles{\@xdyrequiredstyles,#1}}%
\else
  \newcommand*\GlsAddXdyStyle[1]{%
    \glsnxindywarning\GlsAddXdyStyle}
\fi
\ifglxindy
  \newcommand*\GlsSetXdyStyles[1]{%
    \edef\@xdyrequiredstyles{#1}}
\else
  \newcommand*\GlsSetXdyStyles[1]{%
    \glsnxindywarning\GlsSetXdyStyles}
\fi

```

```

\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
    \ifglxindy@glxnumbers
      \def\@xdylettergroups{(define-letter-group
        \string"glxnumbers\string"^^J\space\space\space
        :prefixes (\string"0\string" \string"1\string"
          \string"2\string" \string"3\string" \string"4\string"
          \string"5\string" \string"6\string" \string"7\string"
          \string"8\string" \string"9\string")^^J\space\space\space
          \@xdynumbergrouporder)}
    \else
      \def\@xdylettergroups{}
    \fi
  \fi
  \newcommand*\GlsAddLetterGroup[2]{%
    \expandafter\toks@\expandafter{\@xdylettergroups}%
    \protected@edef\@xdylettergroups{\the\toks@^^J%
      (define-letter-group \string"#1\string"^^J\space\space\space#2)}%
    }%
  \newcommand*\forallglossaries[3][\@glo@types]{%
    \@for#2:=#1\do{\ifx#2\@empty\else#3\fi}%
  }

  \newcommand*\forallacronyms[2]{%

```

```

\@for#1:=\@glsacronymlists\do{\ifx#1\@empty\else#2\fi}%
}
\newcommand*\forlgsentries}[3][\glsdefaulttype]{%
\protected@edef\@glo@list{\csname glolist@#1\endcsname}%
\@for#2:=\@glo@list\do
{%
\ifdefempty{#2}{\#3}%
}%
}
\newcommand*\forallglsentries}[3][\@glo@types]{%
\expandafter\forallglossaries\expandafter[#1]{\@this@glo@}%
{%
\forlgsentries[\@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{\ifglentryexists}[3]{%
\ifcsundef{glo@glsdetoklabel{#1}@name}{#3}{#2}%
}
\newcommand*\ifglused}[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@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]{%
\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@glsetoklabel{#1}@symbol}%
  \ifdefempty\@glo@symbol
    {#3}%
  {%
    \ifdefequal\@glo@symbol\@gls@default@value
      {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshaslong}[3]{%
  \letcs{\@glo@long}{glo@glsetoklabel{#1}@long}%
  \ifdefempty\@glo@long
    {#3}%
  {%
    \ifdefequal\@glo@long\@gls@default@value
      {#3}%
    {#2}%
  }%
}
\newrobustcmd*{\ifglshasshort}[3]{%
  \letcs{\@glo@short}{glo@glsetoklabel{#1}@short}%
  \ifdefempty\@glo@short
    {#3}%
  {%
    \ifdefequal\@glo@short\@gls@default@value
      {#3}%
    {#2}%
  }%
}
\newcommand{\ifglsfieldvoid}[4]{\ifcsvoid{glo@glsetoklabel{#2}@#1}{#3}{#4}}
\newrobustcmd*{\ifglshasfield}[4]{%
  \glsdoifexists{#2}%
  {%
    \letcs{\@glo@thisvalue}{glo@glsetoklabel{#2}@#1}%
    \ifdef\@glo@thisvalue
      {%
        \ifdefempty\@glo@thisvalue
          {%
            #4%
          }%
        {%
          \ifdefequal\@glo@thisvalue\@gls@default@value
            {%
              #4%
            }%
          {%
            \let\glscurrentfieldvalue\@glo@thisvalue
            #3%
          }%
        }%
      }%
    }%
  }%
}

```

```

}%
}%
{%
  \@gls@fetchfield{\@gls@thisfield}{#1}%
  \ifdef\@gls@thisfield
  {%
    \letcs{\@glo@thisvalue}{glo\@glsdetoklabel{#2}\@gls@thisfield}%
    \ifdefempty\@glo@thisvalue
    {%
      #4%
    }%
  }%
  {%
    \ifdefequal\@glo@thisvalue\@gls@default@value
    {%
      #4%
    }%
    {%
      \let\glscurrentfieldvalue\@glo@thisvalue
      #3%
    }%
  }%
}%
{%
  \GlossariesWarning{Unknown entry field ‘#1’}%
  #4%
}%
}%
}%
}

\newcommand*{\glscurrentfieldvalue}{}
\newcommand*{\@glo@types}{,}
\newcommand*{\@gls@provide@newglossary{%
  \protected@write\@auxout{}{\string\providecommand\string\@newglossary[4]{}}%
  \let\@gls@provide@newglossary\relax
}

\newcommand*{\@gls@defaulttype}{\glsdefaulttype}
\newcommand*{\defglsentryfmt}[2][\glsdefaulttype]{%
  \protected@edef\@glo@tmp{#1}%
  \ifx\@glo@tmp\@gls@defaulttype
    \GlossariesWarning{\string\defglsentryfmt\space attempting to set
      a display style for \string\glsdefaulttype\space but there
      doesn't seem to be a default glossary. If you have used
      'nomain', you need to define an alternative glossary}%
  \else
    \csgdef{gls@\@glo@tmp @entryfmt}{#2}%
  \fi
}

\newcommand*{\glsdoentryfmt}[1]{\csuse{gls@#1@entryfmt}}
\newcommand*{\@gls@forbidtexext}[1]{%
  \ifbool{expr}{test}{\ifdefstring{#1}{tex}}

```

```

        or test {\ifdefstring{#1}{TEX}}
    {%
      \def#1{nottex}%
      \PackageError{glossaries}%
        {Forbidden '.tex' extension replaced with '.nottex'}%
        {I'm sorry, I can't allow you to do something so reckless.\MessageBreak
        Don't use '.tex' as an extension for a temporary file.}%
    }%
  {%
  }%
}
\newcommand*{\gls@gobbleopt}{\new@ifnextchar[{\@gls@gobbleopt}{}]}
\def\@gls@gobbleopt[#1]{
\newcommand*{\newglossary}{\@ifstar\s@newglossary\ns@newglossary}
\newcommand*{\s@newglossary}[2]{%
  \ns@newglossary[#1-glg]{#1}{#1-gls}{#1-glo}{#2}%
}
\newcommand*{\ns@newglossary}[5][glg]{%
  \doifglossarynoexistsordo{#2}%
  {%
    \ifundef\glsdefaulttype
    {%
      \gdef\glsdefaulttype{#2}%
    }{%
      \toks@{#2}\protected@edef\@glo@types{\@glo@types\the\toks@,}%
      \expandafter\gdef\csname glolist@#2\endcsname{,}%
      \expandafter\edef\csname @glotype@#2@log\endcsname{#1}%
      \expandafter\edef\csname @glotype@#2@in\endcsname{#3}%
      \expandafter\edef\csname @glotype@#2@out\endcsname{#4}%
      \expandafter\@gls@forbidtexext\csname @glotype@#2@log\endcsname
      \expandafter\@gls@forbidtexext\csname @glotype@#2@in\endcsname
      \expandafter\@gls@forbidtexext\csname @glotype@#2@out\endcsname
      \expandafter\def\csname @glotype@#2@title\endcsname{#5}%
      \@gls@provide@newglossary
      \protected@write\@auxout{\string\@newglossary{#2}{#1}{#3}{#4}}%
      \ifcsundef{gls@#2@entryfmt}%
      {%
        \defglsentryfmt[#2]{\glsentryfmt}%
      }%
      {}%
      \@gls@defsortcount{#2}%
      \@ifnextchar[{\@gls@setcounter{#2}}%
        {\@gls@setcounter{#2}[\glscounter]}%
    }%
  }%
  {%
  \gls@gobbleopt
  }%
}
\newcommand*{\altnewglossary}[3]{%
  \newglossary[#2-glg]{#1}{#2-gls}{#2-glo}{#3}%

```

```

}
\@onlypreamble{\newglossary}
\@onlypremakeg\newglossary
\newcommand*{\@newglossary}[4]{%
\def\@gls@setcounter#1[#2]{%
  \expandafter\def\csname @gls@#1@counter\endcsname{#2}%
  \ifglxindy
    \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
  {%
    \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}%
\ifstrempy{#3}{\def#2{}}%
{%
\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*{\@glsnodelsc{%
  \PackageError{glossaries}
  {%
    description key required in \string\newglossaryentry\space
    for entry '@glo@label'%
  }%
  {%
    You haven't specified the entry description%
  }%
}%
\newcommand*{\@gls@missingnumberlist}[1]{%
  ??%
  \ifglssavenumberlist
    \GlossariesWarning{Missing number list for entry '#1'}.

```

```

    Maybe makeglossaries + rerun required}%
\else
  \PackageError{glossaries}%
  {Package option 'savenumberlist=true' required}%
  {%
    You must use the 'savenumberlist' package option
    to reference location lists.%
  }%
\fi
}
\newcommand*{\@glsdefaultsort}{\@glo@name}
\newcount\gls@level
\newcommand{\@gls@noexpand@field}[3]{%
\expandafter\global\expandafter
  \let\csname glo@#1@#2\endcsname#3%
}
\newcommand{\@gls@noexpand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \edef\@gls@value{\expandonce{#1}}%
    \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}%
}%
\ifdefequal{#4}{\@gls@default@value}%
{%
  \edef\@gls@value{\expandonce{#1}}%
  \@gls@noexpand@field{#2}{#3}{\@gls@value}%
}%
{%
  \@gls@noexpand@field{#2}{#3}{#4}%
}%
}%
}
\newcommand{\@gls@expand@field}[3]{%
\expandafter
  \protected@xdef\csname glo@#1@#2\endcsname{#3}%
}
\newcommand{\@gls@expand@fields}[4]{%
\ifcsdef{gls@assign@#3@field}
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \edef\@gls@value{\expandonce{#1}}%
    \csuse{gls@assign@#3@field}{#2}{\@gls@value}%
  }%

```



```

}%
{%
  \expandafter\@gls@startswithexpandonce#4\relax\relax\gls@endcheck
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
  {%
    \csuse{gls@assign@#3@field}{#2}{#4}%
  }%
}%
}%
{%
  \ifdefequal{#4}{\@gls@default@value}%
  {%
    \@@gls@expand@field{#2}{#3}{#1}%
  }%
  {%
    \@@gls@expand@field{#2}{#3}{#4}%
  }%
}%
}
\def\@gls@expandonce{\expandonce}
\def\@gls@startswithexpandonce#1#2\gls@endcheck#3#4{%
  \def\@gls@tmp{#1}%
  \ifdefequal{\@gls@expandonce}{\@gls@tmp}{#3}{#4}%
}
\let\gls@assign@field\@gls@expand@fields
\newcommand*{\glsexpandfields}{%
  \let\gls@assign@field\@gls@expand@fields
}
\newcommand*{\glsnoexpandfields}{%
  \let\gls@assign@field\@gls@noexpand@fields
}
\newrobustcmd{\newglossaryentry}[2]{%
  \glsdoifnoexists{#1}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\newcommand*{\gls@defdocnewglossaryentry}{%
  \let\newglossaryentry\new@glossaryentry
}
\newrobustcmd{\provideglossaryentry}[2]{%
  \ifglsentryexists{#1}%
  {}%
  {%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\@onlypreamble{\provideglossaryentry}

```

```

\newcommand*{\gls@glsdefs@labellist}{%
\newcommand*{\gls@glsdefs@addlabel}[1]{\listxadd{\gls@glsdefs@labellist}{#1}}
\newcommand*{\gls@docdef@labellist}{%
\newcommand*{\gls@docdef@addlabel}[1]{\listxadd{\gls@docdef@labellist}{#1}}
\newrobustcmd{\new@glossaryentry}[2]{%
\ifundef\@gls@deffile
{%
\global\newwrite\@gls@deffile
\immediate\openout\@gls@deffile=\jobname.glsdefs
}%
}%
\ifglsentryexists{#1}%
{%
\xifinlist{#1}{\gls@docdef@labellist}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in the
document environment (ignoring new definition)}%
}%
{%
\xifinlist{#1}{\gls@glsdefs@labellist}%
{%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
{%
\GlossariesWarning{Entry ‘#1’ has already been defined in
the preamble (in which case remove the duplicate definition)
or you have just updated to glossaries.sty
v4.47 (in which case delete ‘\jobname.glsdefs’)}%
}%
}%
}%
\gls@defglossaryentry{#1}{#2}%
\gls@docdef@addlabel{#1}%
\@gls@writedef{#1}%
}%
}
\AtBeginDocument{\gls@begindocdefs}
\AtEndDocument{\ifdef\@gls@deffile{\closeout\@gls@deffile}{}}
\newcommand*{\gls@begindocdefs}{%
\@gls@enablesavenonumberlist
\edef\@gls@restoreat{\noexpand\catcode‘\noexpand\@=\number\catcode‘\@‘\relax}%
\makeatletter
\InputIfFileExists{\jobname.glsdefs}{%}{%
\@gls@restoreat
\undef\@gls@restoreat
\gls@defdocnewglossaryentry
}

```

```

\newcommand*{\glsdefs@newdocentry}[2]{%
  \ifglentryexists{#1}%
  {\GlossariesWarning{entry ‘#1’ has already been defined in the preamble
    (if you have moved the definition there from the document
    environment, delete \jobname.glsdefs)}}%
  {%
    \gls@glsdefs@addlabel{#1}%
    \gls@defglossaryentry{#1}{#2}%
  }%
}
\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{\glsfieldxddef}[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{\glsfieldddef}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%
      \expandafter\def\csname glo@\@glo@label @#2\endcsname{#3}%
    }%
    {%
      \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
  }%
}
\newrobustcmd{\glsfieldfetch}[3]{%
  \glsdoifexists{#1}%
  {%
    \protected@edef\@glo@label{\glsdetoklabel{#1}}%
    \ifcsdef{glo@\@glo@label @#2}%
    {%

```

```

        \letcs#3{glo@\@glo@label @#2}%
    }%
    {%
        \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
    }%
}%
}
\newcommand*{\glsunexpandedfieldvalue}[2]{%
    \ifcsname glo@\glsdetoklabel{#1}@#2\endcsname
        \expandafter\expandonce\csname glo@\glsdetoklabel{#1}@#2\expandafter\endcsname
    \fi
}
\newrobustcmd{\ifglsfieldeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstring{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newrobustcmd{\ifglsfielddefeq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \expandafter\ifdefstrequal\csname glo@\@glo@label @#2\endcsname{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Field ‘#2’ doesn’t exist}{}%
        }%
    }%
}
\newcommand{\ifglsfieldcseq}[5]{%
    \glsdoifexists{#1}%
    {%
        \protected@edef\@glo@label{\glsdetoklabel{#1}}%
        \ifcsdef{glo@\@glo@label @#2}%
        {%
            \ifcsstrequal{glo@\@glo@label @#2}{#3}{#4}{#5}%
        }%
        {%
            \PackageError{glossaries}{Key ‘#2’ doesn’t exist}{}%
        }%
    }%
}

```



```

}
\newcommand*{\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]{%
  \ifglstryexists{#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@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{}%
  }%

```

```

}%
{%
\ifglentryexists{\@glo@parent}%
{%
\expandafter\protected@xdef\csname glo@\@glo@label @parent\endcsname{%
\@glo@parent}%
\gls@level=\csname glo@\@glo@parent @level\endcsname\relax
\advance\gls@level by 1\relax
\ifx\@glo@name\@glsnoname
\expandafter\let\expandafter\@glo@name
\csname glo@\@glo@parent @name\endcsname
\ifx\@glo@plural\@gls@default@value
\expandafter\let\expandafter\@glo@plural
\csname glo@\@glo@parent @plural\endcsname
\fi
\fi
}%
{%
\PackageError{glossaries}%
{%
Invalid parent '\@glo@parent'
for entry '\@glo@label' - parent doesn't exist%
}%
{%
Parent entries must be defined before their children%
}%
\def\@glo@parent{}%
\expandafter\gdef\csname glo@\@glo@label @parent\endcsname{}%
}%
}%
\fi
\expandafter\xdef\csname glo@\@glo@label @level\endcsname{\number\gls@level}%
\gls@assign@field{\@glo@name}{\@glo@label}{sortvalue}{\@glo@sort}%
\letcs\@glo@sort{glo@\@glo@label @sortvalue}%
\gls@assign@field{\@glo@name}{\@glo@label}{text}{\@glo@text}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname\glspluralsuffix}%
{\@glo@label}{plural}{\@glo@plural}%
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @text\endcsname}%
{\@glo@label}{first}{\@glo@first}%
\ifx\@glo@first\@gls@default@value
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @plural\endcsname}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\else
\expandafter\gls@assign@field\expandafter
{\csname glo@\@glo@label @first\endcsname\glspluralsuffix}%
{\@glo@label}{firstpl}{\@glo@firstplural}%
\fi

```

```

\ifcsundef{@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
  \let{@glo@name}@gls@name
\fi
\gls@assign@field{}{\@glo@label}{name}{\@glo@name}%
\ifcsundef{glo@{@glo@label @numberlist}}%
{%
  \csxdef{glo@{@glo@label @numberlist}}{%
    \noexpand@gls@missingnumberlist{@glo@label}}%
}%
{%
  \let{@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@desc
  \let{@glo@desc}@gls@desc
\fi
\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\@gl@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{\@gl@detoklabel{#1}}%
\protected@edef\@glo@type{\csname glo@\@glo@thislabel @type\endcsname}%
\def\@glo@list{,%
\for@gl@entries[\@glo@type]{\@glo@label}%
{%
\ifdefequal\@glo@thislabel\@glo@label
}{\protected@eappto\@glo@list{\@glo@label,}}%
}%
\cslet\@glolist{\@glo@type}{\@glo@list}%
\csdef{glo@\@glo@thislabel @type}{#2}%
}
\ifgl@xindy
\newcommand*{\@glossaryentryfield}{\string\@glossentry}
\else

```

```

\newcommand*{\@glossaryentryfield}{\string\glossentry}
\fi
\ifglxindy
\newcommand*{\@glossarysubentryfield}{%
\string\subglossentry}
\else
\newcommand*{\@glossarysubentryfield}{%
\string\subglossentry}
\fi
\newcommand{\@glo@storeentry}[1]{%
\protected@edef\@glo@esclabel{#1}%
\@gls@checkmkidxchars\@glo@esclabel
\protected@edef\@glo@sort{\csname glo@#1@sort\endcsname}%
\@gls@checkmkidxchars\@glo@sort
\@gls@checkmkidxchars\@glo@prefix
\protected@edef\@glo@parent{\csname glo@#1@parent\endcsname}%
\ifglxindy
\ifx\@glo@parent\@empty
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
(\string"\@glo@sort\string" %
\string"\@glo@prefix\@glossaryentryfield{\@glo@esclabel}\string") %
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname
(\string"\@glo@sort\string" %
\string"\@glo@prefix\@glossarysubentryfield
{\csname glo@#1@level\endcsname}{\@glo@esclabel}\string") %
}%
\fi
\else
\ifx\@glo@parent\@empty
\@onelevel@sanitize\@glo@prefix
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossaryentryfield{\@glo@esclabel}%
}%
\else
\expandafter\protected@xdef\csname glo@#1@index\endcsname{%
\csname glo@\@glo@parent @index\endcsname\@gls@levelchar
\@glo@sort\@gls@actualchar\@glo@prefix
\@glossarysubentryfield
{\csname glo@#1@level\endcsname}{\@glo@esclabel}%
}%
\fi
\fi
}
\newcommand*{\@gls@if@measuring}[2]{%
\ifmeasuring@
#1%

```

```

\else
  #2%
\fi
}
\let\gls@org@refstepcounter\refstepcounter
\let\gls@org@label\label
\newcommand{\gls@if@measuring}[2]{#2}
\AtBeginDocument{%
  \ifpackageloaded{amsmath}%
  {\let\gls@if@measuring\@gls@if@measuring}%
  }%
  \let\gls@org@refstepcounter\refstepcounter
  \let\gls@org@label\label
}
\newcommand*\gls@ifnotmeasuring[1]{\glsifmeasuring{#1}}
\newcommand*\glsifmeasuring{\gls@if@measuring}
\newcommand*\gls@local@stepcounter[1]{%
  \ifcsdef{c#1}%
  {%
    \epreto\gls@measuring@reset{\noexpand\setcounter{#1}{\number\value{#1}}}%
    \expandafter\advance\csname c#1\endcsname by \@ne
  }%
  }%
}
}
\newcommand{\gls@start@measuring}{%
  \ifx\glsifmeasuring\@firstoftwo
  \else
    \let\gls@org@target\glstarget
    \let\gls@org@link\glslink
    \def\gls@measuring@reset{}%
  \fi
  \let\glsifmeasuring\@firstoftwo
  \let\refstepcounter\gls@local@stepcounter
  \let\label\@gobble
  \let\glstarget\@secondoftwo
  \let\@glslink\@secondoftwo
}
\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{\glsmeasureheight}[2]{%
\settoheight{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasuredepth}[2]{%
\settodepth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\newcommand{\glsmeasurewidth}[2]{%
\settowidth{#1}{\gls@start@measuring#2\gls@stop@measuring}%
}
\def\@gls@patchtabularx#1\hbox#2#3!!{%
\def\TX@trial##1{#1\hbox{\gls@start@measuring#2\gls@stop@measuring}#3}%
}
\newcommand*\glspatchtabularx{%
\ifdef\TX@trial
{%
\expandafter\@gls@patchtabularx\TX@trial{##1}!!%
\let\glspatchtabularx\relax
}%
}%
}
\newcommand*\{glsreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsreset{#1}%
}%
}%
}
\newcommand*\{glslocalreset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glslocalreset{#1}%
}%
}%
}
\newcommand*\{glsunset}[1]{%
\gls@ifnotmeasuring
{%
\glsdoifexists{#1}%
{%
\@glsunset{#1}%
}%
}%
}
}

```



```

\newcommand*\glslocalunset}[1]{%
  \gls@ifnotmeasuring
  {%
    \glsdoifexists{#1}%
    {%
      \glslocalunset{#1}%
    }%
  }%
}
\newcommand*\@glslocalunset{\@glslocalunset}
\newcommand*\@glslocalunset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iftrue
}
\newcommand*\@glsunset{\@glsunset}
\newcommand*\@glsunset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagtrue\endcsname
}
\newcommand*\@glslocalreset{\@glslocalreset}
\newcommand*\@glslocalreset}[1]{%
  \expandafter\let\csname ifglo@glsdetoklabel{#1}@flag\endcsname\iffalse
}
\newcommand*\@glsreset{\@glsreset}
\newcommand*\@glsreset}[1]{%
  \expandafter\global\csname glo@glsdetoklabel{#1}@flagfalse\endcsname
}
\newcommand*\glsresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsreset{\@glsentry}%
  }%
}
\newcommand*\glslocalresetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalreset{\@glsentry}%
  }%
}
\newcommand*\glsunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glsunset{\@glsentry}%
  }%
}
\newcommand*\glslocalunsetall}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glsentry}%
  {%
    \glslocalunset{\@glsentry}%
  }%
}
\newcommand*\@newglossaryentry@defcounters{%

```

```

\csdef{glo@%glo@label @currcount}{0}%
\csdef{glo@%glo@label @prevcount}{0}%
}
\newif\ifglsresetcurrcount
\glsresetcurrcountfalse
\newcommand*{\glsenableentrycount}{%
\let\@newglossaryentry@defcounters\@newglossaryentry@defcounters
\renewcommand*{\gls@defdocnewglossaryentry}{%
\renewcommand*newglossaryentry[2]{%
\PackageError{glossaries}{\string\newglossaryentry\space
may only be used in the preamble when entry counting has
been activated}{If you use \string\glsenableentrycount\space
you must place all entry definitions in the preamble not in
the document environment}%
}%
}%
\newcommand*{\glsentrycurrcount}[1]{%
\ifcsundef{glo@%glsdetoklabel{##1}@currcount}%
{0}{\@gls@entry@field{##1}{currcount}}%
}%
\newcommand*{\glsentryprevcount}[1]{%
\ifcsundef{glo@%glsdetoklabel{##1}@prevcount}%
{0}{\@gls@entry@field{##1}{prevcount}}%
}%
\renewcommand*{\@glsunset}[1]{%
\@glsunset{##1}%
\@gls@increment@currcount{##1}%
}%
\renewcommand*{\@glslocalunset}[1]{%
\@glslocalunset{##1}%
\@gls@local@increment@currcount{##1}%
}%
\renewcommand*{\@glsreset}[1]{%
\@glsreset{##1}%
\ifglsresetcurrcount\csgdef{glo@%glsdetoklabel{##1}@currcount}{0}\fi
}%
\renewcommand*{\@glslocalreset}[1]{%
\@glslocalreset{##1}%
\ifglsresetcurrcount\csdef{glo@%glsdetoklabel{##1}@currcount}{0}\fi
}%
\def\@cgls@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax
\cglsformat{##2}{##3}%
\glsunset{##2}%
\else
\@gls@{##1}{##2}[##3]%
\fi
}%
\def\@cglspl@##1##2[##3]{%
\ifnum\glsentryprevcount{##2}=1\relax

```

```

        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cglsp@{##1}{##2}[##3]%
    \fi
}%
\def\@cGls@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglformat{##2}{##3}%
        \glset{##2}%
    \else
        \cGls@{##1}{##2}[##3]%
    \fi
}%
\def\@cGlspl@##1##2[##3]{%
    \ifnum\glsetprevcount{##2}=1\relax
        \cglspformat{##2}{##3}%
        \glset{##2}%
    \else
        \cGlspl@{##1}{##2}[##3]%
    \fi
}%
\AtEndDocument{\@gls@write@entrycounts}%
\renewcommand*{\@gls@entry@count}[2]{%
    \csdef{glo@\glsetoklabel{##1}@prevcount}{##2}%
}%
\let\glsetenableentrycount\relax
}
\@onlypreamble\glsetenableentrycount
\newcommand*{\@gls@increment@currcount}[1]{%
    \csxdef{glo@\glsetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@local@increment@currcount}[1]{%
    \csedef{glo@\glsetoklabel{##1}@currcount}{%
        \number\numexpr\glsetentrycurrcount{##1}+1}%
}
\newcommand*{\@gls@write@entrycounts}{%
    \immediate\write\@auxout
        {\string\providecommand*\string\@gls@entry@count}[2]{}}%
    \forallglsetentries{\@glsetentry}{%
        \ifglsetused{\@glsetentry}%
        {\immediate\write\@auxout
            {\string\@gls@entry@count{\@glsetentry}{\glsetentrycurrcount{\@glsetentry}}}}%
        {}%
    }%
}
\newcommand*{\@gls@entry@count}[2]{%
}
\newrobustcmd*{\cglsp}{\@gls@hyp@opt\@cglsp}
\newcommand*{\@cglsp}[2][{}]{%

```

```

\new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
\GlossariesWarning{\string\cgl\space is defaulting to
\string\gl\space since you haven't enabled entry counting}%
\@cgl@{#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
\ifglshaslong{#1}{\glentrylong{#1}}{\glentryfirst{#1}}#2%
}
\newrobustcmd*{\cgl}{\@cgl@hyp@opt\@cgl}
\glsmfuaddmap{\cgl}{\cgl}
\newcommand*{\@cgl}[2][]{%
\new@ifnextchar[{\@cgl@{#1}{#2}}{\@cgl@{#1}{#2}[]}%
}
\def\@cgl@#1#2[#3]{%
\GlossariesWarning{\string\cgl\space is defaulting to
\string\gl\space since you haven't enabled entry counting}%
\@cgl@{#1}{#2}[#3]%
}
\newcommand*{\cglformat}[2]{%
\ifglshaslong{#1}{\glentrylong{#1}}{\glentryfirst{#1}}#2%
}
\newrobustcmd*{\cglpl}{\@cgl@hyp@opt\@cglpl}
\glsmfuaddmap{\cglpl}{\cglpl}
\newcommand*{\@cglpl}[2][]{%
\new@ifnextchar[{\@cglpl@{#1}{#2}}{\@cglpl@{#1}{#2}[]}%
}
\def\@cglpl@#1#2[#3]{%
\GlossariesWarning{\string\cglpl\space is defaulting to
\string\glpl\space since you haven't enabled entry counting}%
\@cglpl@{#1}{#2}[#3]%
}
\newcommand*{\cglplformat}[2]{%
\ifglshaslong{#1}{\glentrylongpl{#1}}{\glentryfirstplural{#1}}#2%
}
\newrobustcmd*{\cglpl}{\@cglpl@hyp@opt\@cglpl}
\glsmfuaddmap{\cglpl}{\cglpl}
\newcommand*{\@cglpl}[2][]{%
\new@ifnextchar[{\@cglpl@{#1}{#2}}{\@cglpl@{#1}{#2}[]}%
}
\def\@cglpl@#1#2[#3]{%
\GlossariesWarning{\string\cglpl\space is defaulting to
\string\glpl\space since you haven't enabled entry counting}%
\@cglpl@{#1}{#2}[#3]%
}
\newcommand*{\cglplformat}[2]{%
\ifglshaslong{#1}{\glentrylongpl{#1}}{\glentryfirstplural{#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
{%
\glsupercase{\glsentryplural{\glslabel}\glsinsert}%
}%
{%
\glsupercase{\glsentryfirstplural{\glslabel}\glsinsert}%
}%
}%
}%
\glscapscase
{%
\ifglsused\glslabel
{%
\glsentrytext{\glslabel}\glsinsert
}%
{%
\glsentryfirst{\glslabel}\glsinsert
}%
}%
}

```

```

}%
{%
  \ifglused\glslabel
  {%
    \Glsentrytext{\glslabel}\glsinsert
  }%
  {%
    \Glsentryfirst{\glslabel}\glsinsert
  }%
}%
{%
  \ifglused\glslabel
  {%
    \glsuppercase{\Glsentrytext{\glslabel}\glsinsert}%
  }%
  {%
    \glsuppercase{\Glsentryfirst{\glslabel}\glsinsert}%
  }%
}%
}%
}%
{%
  \glscustomtext\glsinsert
}%
}
\newcommand*{\glsgenacfmt}{%
  \ifdefempty\glscustomtext
  {%
    \ifglused\glslabel
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert
        }%
        {%
          \glsuppercase
            {\acronymfont{\Glsentryshortpl{\glslabel}}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
      }%
      {%

```

```

        \acronymfont{\Glsentryshort{\glslabel}}\glsinsert
    }%
    {%
        \glsuppercase
        {\acronymfont{\glsentryshort{\glslabel}}\glsinsert}%
    }%
}%
}%
{%
    \glsifplural
    {%
        \glscapscase
        {%
            \genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genplacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \glsuppercase
            {\genplacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
    {%
        \glscapscase
        {%
            \genacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \Genacrfullformat{\glslabel}{\glsinsert}%
        }%
        {%
            \glsuppercase
            {\genacrfullformat{\glslabel}{\glsinsert}}%
        }%
    }%
}%
}%
{%
    \glscustomtext
}%
}
\newcommand*{\genacrfullformat}[2]{%
    \glsentrylong{#1}#2\space
    (\protect\firstacronymfont{\glsentryshort{#1}})%
}
\newcommand*{\Genacrfullformat}[2]{%
    \protected@edef\gls@text{\genacrfullformat{#1}{#2}}%
    \expandafter\glssentencecase\expandafter{\gls@text}%
}

```

```

\newcommand*\genplacrfullformat}[2]{%
  \glsentrylongpl{#1}#2\space
  (\protect\firstacronymfont{\glsentryshortpl{#1}})%
}
\newcommand*\Genplacrfullformat}[2]{%
  \protected@edef\gls@text{\genplacrfullformat{#1}{#2}}%
  \expandafter\glsentencecase\expandafter{\gls@text}%
}
\define@key{glslink}{counter}{%
  \ifcsundef{c@#1}%
  {%
    \PackageError{glossaries}%
    {There is no counter called ‘#1’}%
    {%
      The counter key should have the name of a valid counter
      as its value%
    }%
  }%
  {%
    \def\@gls@counter{#1}%
  }%
}
\define@key{glslink}{format}{%
  \def\@glsnumberformat{#1}}
\newcommand*\@gls@glslink@hyper@update@hook{}
\define@boolkey{glslink}{hyper}[true]{\@gls@glslink@hyper@update@hook}
\ifdef\hyperlink{\KV@glslink@hypertrue}{\KV@glslink@hyperfalse}
\newcommand*\@gls@save@glslocal{%
  \let\if@org@KV@glslink@local\ifKV@glslink@local
}
\newcommand*\@gls@restore@glslocal{%
  \ifKV@glslink@local
    \let\@gls@do@glsunset\glslocalunset
  \else
    \let\@gls@do@glsunset\glsunset
  \fi
  \let\ifKV@glslink@local\if@org@KV@glslink@local
}
\newcommand*\@gls@do@glsunset}[1]{\glsunset{#1}}
\define@boolkey{glslink}{local}[true]{}
\@gls@save@glslocal
\newcommand*\glslinkvar}[3]{#1}
\newcommand*\@gls@hyp@opt}[1]{%
  \let\glslinkvar\@firstofthree
  \let\@gls@hyp@opt@cs#1\relax
  \@ifstar{\s@gls@hyp@opt}%
  {\@ifnextchar+{\@firstoftwo{\p@gls@hyp@opt}}{#1}}%
}
\newcommand*\s@gls@hyp@opt}[1][]{%
  \let\glslinkvar\@secondofthree

```



```

\@gls@hyp@opt@cs[hyper=false,#1]}
\newcommand*{\p@gls@hyp@opt}[1][{}]{%
\let\glslinkvar\@thirdofthree
\@gls@hyp@opt@cs[hyper=true,#1]}
\newrobustcmd*{\glslink}{%
\@gls@hyp@opt\@gls@link
}
\newcommand*{\@gls@link}[3][{}]{%
\glsdoifexistsordo{#2}%
{%
\let\do@gls@link@checkfirsthyper\relax
\@gls@link[#1]{#2}{#3}%
}%
\glstextformat{#3}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glslink}{%
\@gls@hyp@opt\@Gls@link
}
\newcommand*{\@Gls@link}[3][{}]{%
\@gls@link[#1]{#2}{\glsentencecase{#3}}%
}
\glsmfuaddmap{\glslink}{\Glslink}
\newcommand*{\glspostlinkhook}{}
\newcommand*{\@gls@link@checkfirsthyper}{%
\ifglsused{\glslabel}%
{%
}%
{%
\gls@checkisacronymlist\glstype
\ifglshyperfirst
\if@glsisacronymlist
\ifglsacrfootnote
\KV@glslink@hyperfalse
\fi
\fi
\else
\KV@glslink@hyperfalse
\fi
}%
\glslinkcheckfirsthyperhook
}
\newcommand*{\glslinkcheckfirsthyperhook}{}
\newcommand*{\glslinkpostsetkeys}{}
\newcommand{\glsifhyperon}[2]{\ifKV@glslink@hyper#1\else#2\fi}
\newcommand*{\do@glsdisablehyperinlist}{%
\expandafter\DTLifinlist\expandafter{\glstype}{\@gls@nohyperlist}%
{\KV@glslink@hyperfalse}{}%
}

```

```

\newcommand*{\@gls@setdefault@glslink@opts}{%
\def\@gls@link[#1]#2#3{%
  \leavevmode
  \protected@edef\glslabel{\glsdetoklabel{#2}}%
  \def\@gls@link@opts{#1}%
  \let\@gls@link@label\glslabel
  \def\@glsnumberformat{glsnumberformat}%
  \protected@edef\@gls@counter{\csname glo@\glslabel @counter\endcsname}%
  \protected@edef\glstype{\csname glo@\glslabel @type\endcsname}%
  \let\org@ifKV@glslink@hyper@ifKV@glslink@hyper
  \@gls@save@glslocal
  \@gls@setdefault@glslink@opts
  \do@glsdisablehyperinlist
  \do@gls@link@checkfirsthyper
  \setkeys{glslink}{#1}%
  \glslinkpostsetkeys
  \@gls@saveentrycounter
  \@gls@setsort{\glslabel}%
  \glsencapwrcontent{\@do@wrglossary{#2}}%
  \ifKV@glslink@hyper
    \@glslink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \else
    \glsdonohyperlink{\glolinkprefix\glslabel}{\glstextformat{#3}}%
  \fi
  \let\ifKV@glslink@hyper\org@ifKV@glslink@hyper
  \@gls@restore@glslocal
}
\newcommand*{\glsencapwrcontent}[1]{\{#1\}}
\newcommand*{\glolinkprefix}{glo:}
\def\glsentrycounter{\glscounter}%
\newcommand*{\@gls@saveentrycounter}{%
\def\@gls@Hcounter{%
\ifthenelse{equal{\@gls@counter}{equation}}{%
{
\ifcsundef{xatlevel@}%
{%
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
}%
{%
\ifx\xatlevel@\@empty
\edef\theglsentrycounter{\expandafter\noexpand
\csname the\@gls@counter\endcsname}%
\else
\savecounters@
\advance\c@equation by 1\relax
\edef\theglsentrycounter{\csname the\@gls@counter\endcsname}%
\ifcsundef{theH\@gls@counter}%
{%
\def\@gls@Hcounter{\theglsentrycounter}%

```

```

    }%
    {%
        \def\@gls@Hcounter{\csname theH\@gls@counter\endcsname}%
    }%
    \protected@edef\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@suffix{glsnumberformat}%
        \else
            \def\@glo@suffix{#2}%
        \fi
    \else
        \def\@glo@prefix{}\def\@glo@suffix{#1#2}%
    \fi
}

```

```

\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}%
  \ifglsclocations
    \for\@gls@tmp:=\gls@protected@pagefmts\do
    {%
      \edef\@gls@sanitized@tmp{\expandafter\@gobble\string\\ \expandonce\@gls@tmp}%
      \@onelevel@sanitize\@gls@sanitized@tmp
      \edef\gls@dosubst{%
        \noexpand\DTLsubstituteall\noexpand\gls@xdystring
        {\@gls@sanitized@tmp}{\expandonce\@gls@tmp}%
      }%
      \gls@dosubst
    }%
  \fi
  \let#1=\gls@xdystring
}
\newcommand{\@gls@checkmkidxchars}[1]{%
  \ifglsxindy
    \@gls@escbsdq{#1}%
  \else
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescquote#1\@nil""\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescactual#1\@nil"??\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkactual#1\@nil??\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkbar#1\@nil||\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
    \def\@gls@checkedmkidx{}%
    \expandafter\@gls@checkescbar#1\@nil\\|\null
    \expandafter\@gls@updatechecked\@gls@checkedmkidx{#1}%
  \fi
}

```

```

\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
\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
\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@quotearchar\@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@quotearchar\@gls@levelchar\@gls@quotearchar\@gls@levelchar}%
            \def\@gls@checklevel{\@gls@checklevel#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotearchar\@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@quotearchar\@gls@actualchar\@gls@quotearchar\@gls@actualchar}%
            \def\@gls@checkactual{\@gls@checkactual#3\null}%
        \fi
    \else
        \edef\@gls@checkedmkidx{\the\@gls@tmpb\the\toks@
            \@gls@quotearchar\@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}%
\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:nnT
{ \tl_if_exist_p:N \glscurrententrylabel }
{ \bool_not_p:n { \tl_if_empty_p:N \glscurrententrylabel } }
{
\glsenryname \glscurrententrylabel
}
}
\ExplSyntaxOff
\newcommand*\glslabelhypertargetprefix{}
\newcommand*\glslabelhypertargetdefs{%
\let\glossentryname\glsenryname
\let\glossentrydesc\glsenrydesc
\let\glossentrysymbol\glsenrysymbol
\let\Glossentryname\glsenryname
\let\Glossentrydesc\glsenrydesc
\let\Glossentrysymbol\glsenrysymbol
\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\glscapscase\@firstofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%
    \def\@glo@text{\csname gls@glstype @entryfmt\endcsname}%
    \@gls@link[#1]{#2}{\@glo@text}%
    \@gls@do@glsunset{#2}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\Gls}{\@gls@hyp@opt\@Gls}
\glsmfuaddmap{\gls}{\Gls}
\newcommand*{\@Gls}[2][{}]{%
  \new@ifnextchar[{\@Gls@{#1}{#2}}{\@Gls@{#1}{#2}[]}]%
}
\def\@Gls@#1#2[#3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glscustomtext\@empty
    \def\glsinsert{#3}%

```

```

\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\GLS}{\@gls@hyp@opt\@GLS}
\glsmfublocker{\GLS}
\newcommand*{\@GLS}[2][]{%
\new@ifnextchar[{\@GLS@{#1}{#2}}{\@GLS@{#1}{#2}[]}%
}
\def\@GLS@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@secondoftwo
\let\glscapscase\@thirdofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\glspl}{\@gls@hyp@opt\@glspl}
\newcommand*{\@glspl}[2][]{%
\new@ifnextchar[{\@glspl@{#1}{#2}}{\@glspl@{#1}{#2}[]}%
}
\def\@glspl@#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@checkfirsthyper
\let\glsifplural\@firstoftwo
\let\glscapscase\@firstofthree
\let\glscustomtext\@empty
\def\glsinsert{#3}%
\def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
\@gls@link[#1]{#2}{\@glo@text}%
\@gls@do@glsunset{#2}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Glspl}{\@gls@hyp@opt\@Glspl}
\glsmfuaddmap{\glspl}{\Glspl}
\newcommand*{\@Glspl}[2][]{%
\new@ifnextchar[{\@Glspl@{#1}{#2}}{\@Glspl@{#1}{#2}[]}%
}
\def\@Glspl@#1#2[#3]{%
\glsdoifexists{#2}%

```

```

{%
  \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
  \let\gl@sifplural\@firstoftwo
  \let\gl@scapscase\@secondofthree
  \let\gl@customtext\@empty
  \def\gl@insert{#3}%
  \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
  \@gl@link[#1]{#2}{\@glo@text}%
  \@gl@do@gl@unset{#2}%
}%
\gl@postlinkhook
}
\newrobustcmd*{\GLSpl}{\@gl@hyp@opt\@GLSpl}
\glsmfublocker{\GLSpl}
\newcommand*{\@GLSpl}[2][{}]{%
  \new@ifnextchar[{\@GLSpl@{#1}{#2}}{\@GLSpl@{#1}{#2}[]}%
}
\def\@GLSpl@#1#2[#3]{%
  \gl@doifexists{#2}%
  {%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@firstoftwo
    \let\gl@scapscase\@thirdofthree
    \let\gl@customtext\@empty
    \def\gl@insert{#3}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\gl@disp}{\@gl@hyp@opt\@gl@disp}
\newcommand*{\@gl@disp}[3][{}]{%
  \gl@doifexists{#2}{%
    \let\do@gl@link@checkfirsthyper\@gl@link@checkfirsthyper
    \let\gl@sifplural\@secondoftwo
    \let\gl@scapscase\@firstofthree
    \def\gl@customtext{#3}%
    \def\gl@insert{}%
    \def\@glo@text{\csname gls@\glstype @entryfmt\endcsname}%
    \@gl@link[#1]{#2}{\@glo@text}%
    \@gl@do@gl@unset{#2}%
  }%
  \gl@postlinkhook
}
\newrobustcmd*{\Glsdisp}{\@gl@hyp@opt\@Glsdisp}
\newcommand*{\@Glsdisp}[3][{}]{%
  \@gl@disp[#1]{#2}{\gl@sentencecase{#3}}%
}
\glsmfuaddmap{\gl@disp}{\Glsdisp}

```

```

\newcommand*{\@gls@link@nocheckfirsthyper}{}
\newcommand{\@gls@field@link}[3]{%
  \glsdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \@gls@link[#1]{#2}{#3}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\glstext}{\@gls@hyp@opt\@glstext}
\newcommand*{\@glstext}[2][]{%
  \new@ifnextchar[{\@glstext@{#1}{#2}}{\@glstext@{#1}{#2}[]}}
\def\@glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentrytext{#2}#3}%
}
\newrobustcmd*{\GLStext}{\@gls@hyp@opt\@GLStext}
\glsmfublocker{\GLStext}
\newcommand*{\@GLStext}[2][]{%
  \new@ifnextchar[{\@GLStext@{#1}{#2}}{\@GLStext@{#1}{#2}[]}}
\def\@GLStext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentrytext{#2}#3}}%
}
\newrobustcmd*{\Glstext}{\@gls@hyp@opt\@Glstext}
\glsmfuaddmap{\glstext}{\Glstext}
\newcommand*{\@Glstext}[2][]{%
  \new@ifnextchar[{\@Glstext@{#1}{#2}}{\@Glstext@{#1}{#2}[]}}
\def\@Glstext@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentrytext{#2}#3}%
}
\newrobustcmd*{\glsfirst}{\@gls@hyp@opt\@glsfirst}
\newcommand*{\@glsfirst}[2][]{%
  \new@ifnextchar[{\@glsfirst@{#1}{#2}}{\@glsfirst@{#1}{#2}[]}}
\def\@glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirst{#2}#3}%
}
\newrobustcmd*{\Glsfirst}{\@gls@hyp@opt\@Glsfirst}
\glsmfuaddmap{\glsfirst}{\Glsfirst}
\newcommand*{\@Glsfirst}[2][]{%
  \new@ifnextchar[{\@Glsfirst@{#1}{#2}}{\@Glsfirst@{#1}{#2}[]}}
\def\@Glsfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\Glsentryfirst{#2}#3}%
}
\newrobustcmd*{\GLSfirst}{\@gls@hyp@opt\@GLSfirst}
\glsmfublocker{\GLSfirst}
\newcommand*{\@GLSfirst}[2][]{%
  \new@ifnextchar[{\@GLSfirst@{#1}{#2}}{\@GLSfirst@{#1}{#2}[]}}
\def\@GLSfirst@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryfirst{#2}#3}}%
}
\newrobustcmd*{\glsplural}{\@gls@hyp@opt\@glsplural}

```

```

\newcommand*{\@glsplural}[2][\%
  \new@ifnextchar[{\@glsplural@{#1}{#2}}{\@glsplural@{#1}{#2}[]}]
\def\@glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\Glsplural}{\@gls@hyp@opt\@Glsplural}
\glsmfuaddmap{\glsplural}{\Glsplural}
\newcommand*{\@Glsplural}[2][\%
  \new@ifnextchar[{\@Glsplural@{#1}{#2}}{\@Glsplural@{#1}{#2}[]}]
\def\@Glsplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\GLSplural}{\@gls@hyp@opt\@GLSplural}
\glsmfublocker{\GLSplural}
\newcommand*{\@GLSplural}[2][\%
  \new@ifnextchar[{\@GLSplural@{#1}{#2}}{\@GLSplural@{#1}{#2}[]}]
\def\@GLSplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsupercase{\glsentryplural{#2}#3}%
}
\newrobustcmd*{\glsfirstplural}{\@gls@hyp@opt\@glsfirstplural}
\newcommand*{\@glsfirstplural}[2][\%
  \new@ifnextchar[{\@glsfirstplural@{#1}{#2}}{\@glsfirstplural@{#1}{#2}[]}]
\def\@glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\glsentryfirstplural{#2}#3}%
}
\newrobustcmd*{\Glsfirstplural}{\@gls@hyp@opt\@Glsfirstplural}
\glsmfuaddmap{\glsfirstplural}{\Glsfirstplural}
\newcommand*{\@Glsfirstplural}[2][\%
  \new@ifnextchar[{\@Glsfirstplural@{#1}{#2}}{\@Glsfirstplural@{#1}{#2}[]}]
\def\@Glsfirstplural@#1#2[#3]{%
  \@gls@field@link{#1}{#2}{\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*{\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}{\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}{\glssupercase{\glssentryuseri{#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}{\glssentryuserii{#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}{\glssupercase{\glssentryuserii{#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}{\glssentryuseriii{#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}{\glssupercase{\glssentryuseriii{#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}{\glsupercase{\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}{\glsupercase{\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}{\glssupercase{\glssentryuservi{#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]{%
  \glstdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\glssentryshort{#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]{%
  \glstdoifexists{#2}%
  {%
    \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \acronymfont{\Glsentryshort{#2}}#3%
    }%
    \@gls@link{#1}{#2}{\csname gls@glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*{\ACRshort}{\@gls@hyp@opt\@ns@ACRshort}
\glsmfublocker{\ACRshort}
\newcommand*{\@ns@ACRshort}[2] [] {%

```

```

\new@ifnextchar[{\@ACRshort{#1}{#2}}{\@ACRshort{#1}{#2} []}%
}
\def\@ACRshort#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@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{\acronymfont{\gl@entryshort{#2}}#3}%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\gl@postlinkhook
}
\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\gl@scapscase\@firstofthree
\let\gl@insert\@empty
\def\glscustomtext{%
\acronymfont{\gl@entryshortpl{#2}}#3%
}%
\@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
}%
\gl@postlinkhook
}
\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]{%
\glsdoifexists{#2}%
{%
\let\do@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\gl@ifplural\@firstoftwo
\let\gl@scapscase\@secondofthree
\let\gl@insert\@empty
\def\glscustomtext{%

```

```

        \acronymfont{\Glsentryshortpl{#2}}#3%
    }%
    \@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRshortpl}{\@gls@hyp@opt\ns@ACRshortpl}
\glsmfublocker{\ACRshortpl}
\newcommand*{\ns@ACRshortpl}[2][]{%
    \new@ifnextchar[{\@ACRshortpl{#1}{#2}}{\@ACRshortpl{#1}{#2} []}%
}
\def\@ACRshortpl#1#2[#3]{%
    \glsdoifexists{#2}%
    {%
        \let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
        \def\glslabel{#2}%
        \let\glsifplural\@firstoftwo
        \let\glscapscase\@thirdofthree
        \let\glsinsert\@empty
        \def\glscustomtext{%
            \glsupercase{\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\glsifplural\@secondoftwo
    \let\glscapscase\@secondofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glentrylong{#2}#3%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@acrlong{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@secondoftwo
    \let\glscapscase\@thirdofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glssupercase{\glentrylong{#2}#3}%
    }%
    \@gl@link[#1]{#2}{\csname gls@\glstype @entryfmt\endcsname}%
  }%
  \glspostlinkhook
}
\newrobustcmd*\@acrlongpl{\@gl@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@gl@link@checkfirsthyper\@gl@link@nocheckfirsthyper
    \def\glslabel{#2}%
    \let\glsifplural\@firstoftwo
    \let\glscapscase\@firstofthree
    \let\glsinsert\@empty
    \def\glscustomtext{%
      \glentrylongpl{#2}#3%
    }%
  }%
}

```



```

\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\Acrlongpl}{\@gls@hyp@opt\@ns@Acrlongpl}
\glsmfuaddmap{\Acrlongpl}{\Acrlongpl}
\newcommand*{\ns@Acrlongpl}[2][]{%
\new@ifnextchar[{\@Acrlongpl{#1}{#2}}{\@Acrlongpl{#1}{#2}[]}%
}
\def\@Acrlongpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@secondofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\Glsentrylongpl{#2}#3%
}%
\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newrobustcmd*{\ACRlongpl}{\@gls@hyp@opt\@ns@ACRlongpl}
\glsmfublocker{\ACRlongpl}
\newcommand*{\ns@ACRlongpl}[2][]{%
\new@ifnextchar[{\@ACRlongpl{#1}{#2}}{\@ACRlongpl{#1}{#2}[]}%
}
\def\@ACRlongpl#1#2[#3]{%
\glsdoifexists{#2}%
{%
\let\do@gls@link@checkfirsthyper\@gls@link@nocheckfirsthyper
\def\glslabel{#2}%
\let\glsifplural\@firstoftwo
\let\glscapscase\@thirdofthree
\let\glsinsert\@empty
\def\glscustomtext{%
\glsuppercase{\Glsentrylongpl{#2}#3}%
}%
\@gls@link[#1]{#2}{\csname gls@glstype @entryfmt\endcsname}%
}%
\glspostlinkhook
}
\newcommand*{\@gls@entry@field}[2]{%
\csuse{glo@glsdetoklabel{#1}@#2}%
}
\newcommand*{\glsletentryfield}[3]{%
\letcs{#1}{glo@glsdetoklabel{#2}@#3}%
}

```

```

\newrobustcmd*{\@Gls@entry@field}[2]{%
  \glsdoifexistsordo{#1}%
  {%
    \letcs\@glo@text{glo@\glsdetoklabel{#1}@#2}%
    \ifdef\@glo@text
    {%
      \expandafter\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{\@Gls@entry@field}[2]{%
  \glstexpdfstring{\@Gls@entry@field{#1}{#2}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{#2}}}%
}
\ifdef\texpdfstring
{\newcommand{\glstexpdfstring}{\texpdfstring}}
{\newcommand{\glstexpdfstring}[2]{#1}}
\newcommand{\Gls@entryname}[1]{\@Gls@entry@field{#1}{name}}
\newcommand*{\Gls@entryname}[1]{%
  \glstexpdfstring{\protect\@Gls@entryname{#1}}%
  {\MFUsentencecase{\@Gls@entry@field{#1}{name}}}%
}
\glsmfuexcl{\Gls@entryname}
\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\glstentrylong\relax

```

```

\expandafter\Glsentrylong\@gls@rest
\else
\expandafter\ifx\@gls@body\glsentryshort\relax
\expandafter\Glsentryshort\@gls@rest
\else
\expandafter\ifx\@gls@body\acronymfont\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\ifx\@gls@body\glsshortaccessdisplay\relax
{%
\let\glsentryshort\Glsentryshort
\@glo@text
}%
\else
\expandafter\glssentencecase\expandafter{\@glo@text}%
\fi
\fi
\fi
\fi
}%
{%
\@Gls@entry@field{#1}{name}%
}%
}
\newcommand*{\glsentrydesc}[1]{\@Gls@entry@field{#1}{desc}}
\newcommand*{\Glsentrydesc}[1]{\@Glsentryfield{#1}{desc}}
\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*{\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}%
}
\glsmfuexcl{\Glsentryuseri}
\newcommand*{\glsentryuserii}[1]{%
    \@gls@entry@field{#1}{userii}%
}
\newcommand*{\Glsentryuserii}[1]{%
    \@Glsentryfield{#1}{userii}%
}
\glsmfuexcl{\Glsentryuserii}
\newcommand*{\glsentryuseriii}[1]{%
    \@gls@entry@field{#1}{useriii}%
}
\newcommand*{\Glsentryuseriii}[1]{%
    \@Glsentryfield{#1}{useriii}%
}
\glsmfuexcl{\Glsentryuseriii}
\newcommand*{\glsentryuseriv}[1]{%
    \@gls@entry@field{#1}{useriv}%
}
\newcommand*{\Glsentryuseriv}[1]{%
    \@Glsentryfield{#1}{useriv}%
}
\glsmfuexcl{\Glsentryuseriv}
\newcommand*{\glsentryuserv}[1]{%
    \@gls@entry@field{#1}{userv}%
}
\newcommand*{\Glsentryuserv}[1]{%
    \@Glsentryfield{#1}{userv}%
}
\glsmfuexcl{\Glsentryuserv}
\newcommand*{\glsentryuservi}[1]{%
    \@gls@entry@field{#1}{uservi}%
}
\newcommand*{\Glsentryuservi}[1]{%
    \@Glsentryfield{#1}{uservi}%
}
\glsmfuexcl{\Glsentryuservi}
\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*{\glentrylong}[1]{\@gls@entry@field{#1}{long}}
\newcommand*{\Glsentrylong}[1]{%
  \@Glsentryfield{#1}{long}%
}
\glsmfuexcl{\Glsentrylong}
\newcommand*{\glentrylongpl}[1]{\@gls@entry@field{#1}{longpl}}
\newcommand*{\Glsentrylongpl}[1]{%
  \@Glsentryfield{#1}{longpl}%
}
\glsmfuexcl{\Glsentrylongpl}
\newcommand*{\glentryfull}[1]{%
  \glentrylong{#1} (\glentryshort{#1})%
}
\newcommand*{\Glsentryfull}[1]{%
  \Glsentrylong{#1} (\glentryshort{#1})%
}
\glsmfuexcl{\Glsentryfull}
\newcommand*{\glentryfullpl}[1]{%
  \acrfullformat{\glentrylongpl{#1}}{\acronymfont{\glentryshortpl{#1}}}%
}
\newcommand*{\Glsentryfullpl}[1]{%
  \Glsentrylongpl{#1} (\glentryshortpl{#1})%
}
\glsmfuexcl{\Glsentryfullpl}
\newcommand*{\glentrynumberlist}[1]{%
  \glsdoifexists{#1}%
  {%
    \@gls@entry@field{#1}{numberlist}%
  }%
}
\@ifpackageloaded{hyperref} {%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \GlossariesWarning
    {%
      \string\glsdisplaynumberlist\space
      doesn't work with hyperref.^^JUsing
      \string\glentrynumberlist\space instead%
    }%
    \glentrynumberlist{#1}%
  }%
}%
{\%
  \newcommand*{\glsdisplaynumberlist}[1]{%
    \glsdoifexists{#1}%
    {%
      \bgroup
      \protected@edef\glo@label{\glsdetoklabel{#1}}%
      \let\@org@glnumberformat\glnumberformat
      \def\glnumberformat##1{##1}%
    }
  }
}

```

```

\protected@edef\the@numberlist{%
  \csname glo@\glo@label @numberlist\endcsname}%
\def\@gls@numlist@sep{%
\def\@gls@numlist@nextsep{%
\def\@gls@numlist@lastsep{%
\def\@gls@thislist{%
\def\@gls@donext@def{%
\renewcommand\do[1]{%
  \protected@edef\@gls@thislist{%
    \@gls@thislist
    \noexpand\@gls@numlist@sep
    ##1%
  }%
  \let\@gls@numlist@sep\@gls@numlist@nextsep
  \def\@gls@numlist@nextsep{\glsnumlistsep}%
  \@gls@donext@def
  \def\@gls@donext@def{%
    \def\@gls@numlist@lastsep{\glsnumlistlastsep}%
  }%
}%
\expandafter \glsnumlistparser \expandafter{\the@numberlist}%
\let\@gls@numlist@sep\@gls@numlist@lastsep
\@gls@thislist
\egroup
}%
}
}
\newcommand*{\glsnumlistsep}{, }
\newcommand*{\glsnumlistlastsep}{ \& }
\newcommand*{\glshyperlink}[2][\glsentrytext{\@glo@label}]{%
  \def\@glo@label{#2}%
  \@glslink{\glo@linkprefix\glsdetoklabel{#2}}{#1}}
\define@key{glossadd}{counter}{\def\@gls@counter{#1}}
\define@key{glossadd}{format}{\def\@glsnumberformat{#1}}
\define@key{glossadd}{types}{\def\@glo@type{#1}}
\newrobustcmd*{\glsadd}[2][{}]{%
  \@gls@adjustmode
  \glsdoifexists{#2}%
  {%
    \def\@glsnumberformat{glsnumberformat}%
    \protected@edef\@gls@counter{\csname glo\glsdetoklabel{#2}@counter\endcsname}%
    \setkeys{glossadd}{#1}%
    \@gls@saveentrycounter
    \@gls@setsort{#2}%
    \@do@wrglossary{#2}%
  }%
}
\glsmfuexcl{\glsadd}
\newcommand*{\@gls@adjustmode}{%
\AtBeginDocument{\renewcommand*{\@gls@adjustmode}{\ifvmode\mbox{}\fi}}

```

```

\newrobustcmd*{\glsaddall}[1][\@gls@types]{%
  \protected@edef\@glo@type{\@glo@types}%
  \setkeys{glossadd}{#1}%
  \forallglsentries[\@glo@type]{\@glo@entry}{%
    \glsadd[#1]{\@glo@entry}%
  }%
}
\newrobustcmd*{\glsaddallunused}[1][\@glo@types]{%
  \forallglsentries[#1]{\@glo@entry}%
  {%
    \ifglsused{\@glo@entry}{\@glsadd[format=glsignore]{\@glo@entry}}%
  }%
}
\newcommand*{\glsignore}[1]{%
  \edef\glsopenbrace{\expandafter\@gobble\string\{ }
  \edef\glsclosebrace{\expandafter\@gobble\string\} }
  \edef\glsbackslash{\expandafter\@gobble\string\ }
  \edef\glsquote#1{\string"#1\string"}
  \edef\glspercentchar{\expandafter\@gobble\string\% }
  \edef\glstildechar{\string~}
  \ifglsxindy
    \newcommand*{\@glsfirstletter}{A}
  \fi
  \newcommand*{\GlsSetXdyFirstLetterAfterDigits}{%
    \@ifstar\s@GlsSetXdyFirstLetterAfterDigits\@GlsSetXdyFirstLetterAfterDigits}
  \ifglsxindy
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}}
    \newcommand*{\s@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \renewcommand*{\@glsfirstletter}{#1}%
      \@onelevel@sanitize\@glsfirstletter
    }
  \else
    \newcommand*{\@GlsSetXdyFirstLetterAfterDigits}[1]{%
      \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}{}%
\@glscataddpredefinedattributes
\openout\glswrite=\istfilename
\write\glswrite{;; xindy style file created by the glossaries
package}%
\write\glswrite{;; for document '\jobname' on
\the\year-\the\month-\the\day}%
\write\glswrite{^^J; required styles^^J}
\@for\@xdystyle:=\@xdyrequiredstyles\do{%
\ifx\@xdystyle\@empty
\else
\protected@write\glswrite{}{(require
\string"\@xdystyle.xdy\string")}%
\fi
}%
\write\glswrite{^^J%
; list of allowed attributes (number formats)^^J}%
\write\glswrite{(define-attributes ((\@xdyattributes)))}%
\write\glswrite{^^J; user defined alphabets^^J}%
\write\glswrite{\@xdyuseralphabets}%
\write\glswrite{^^J; location class definitions^^J}%
\@for\@glsclassI:=\@glscatxdy@locationlist\do{%
\protected@write\glswrite{}{(define-location-class
\string"\@glscatxdy@locationlist\string"^^J\space\space\space
(
:sep "{ }{"
\csname @glscatxdy@Lclass@\@glscatxdy@locationlist\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{\glspcrcentchar\space makeindex style file
created by the glossaries package}
\write\glswrite{\glspcrcentchar\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}
\write\glswrite{delim_n \string\string\delimN \string}
\write\glswrite{delim_r \string\string\delimR \string}
\write\glswrite{headings_flag 1}
\write\glswrite{heading_prefix
    \string\string\glsgroupheading\string\{\string}
\write\glswrite{heading_suffix
    \string\string\}\string\relax
    \string\glresetentrylist \string}
\write\glswrite{symhead_positive \string"glssymbols\string"}
\write\glswrite{numhead_positive \string"glnumbers\string"}
\write\glswrite{page_compositor \string"glscpositor\string"}
\@gls@escbsdq\gls@suffixF
\@gls@escbsdq\gls@suffixFF
\ifx\gls@suffixF\@empty
\else
    \write\glswrite{suffix_2p \string"\gls@suffixF\string"}
\fi
\ifx\gls@suffixFF\@empty
\else
    \write\glswrite{suffix_3p \string"\gls@suffixFF\string"}
\fi
\@gls@writeisthook
\closeout\glswrite
\let\writeist\relax
\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####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@checkescquote####1\noexpand\@nil
\expandonce{\csname#1\endcsname}\expandonce{\csname#1\endcsname}%
\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%
\noexpand\expandafter\noexpand\@gls@checkescactual####1\noexpand\@nil
\noexpand\?\noexpand\?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}}%

```

```

\noexpand\expandafter\noexpand\@gls@checkactual####1\noexpand\@nil
\noexpand?\noexpand?\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkbar####1\noexpand\@nil
\noexpand|\noexpand|\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checkesbar####1\noexpand\@nil
\noexpand||\noexpand||\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\noexpand\def\noexpand\@gls@checkedmkidx{}%
\noexpand\expandafter\noexpand\@gls@checklevel####1\noexpand\@nil
\noexpand!\noexpand!\noexpand\null
\noexpand\expandafter\noexpand\@gls@updatechecked
\noexpand\@gls@checkedmkidx{####1}%
\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
\noexpand\@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\@gls@openfile\csname glo@#1@file\endcsname{#1}%
    \fi
    \@gls@renewglossary
    \writeist
  }%
  {%

```



```

\PackageError{glossaries}%
{Glossary type ‘#1’ not defined}%
{New glossaries must be defined before using \string\makeglossaries}%
}%
}
\newcommand*{\@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}}%
\@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{-l }}%
{\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
\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}}%
                \@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{-l }}%
                {\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
\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}%
{\@gls@run@unrestricted@shell{\@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@onlypreamble\makeg
\let\gls@checkseeallowed\relax
\let\warn@nomakeglossaries\relax
\let\warn@noprintglossary\@makeglossaries@warn@noprintglossary
\iflssavenumberlist
  \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@noref@warn}[1]{%
      \GlossariesWarning{Empty glossary for
        \string\printnoidxglossary[type={##1}].
        Rerun may be required (or you may have forgotten to use
        commands like \string\gls)}%
    }%
    \let\@gls@checkmkidxchars\@gobble
    \glsesclocationsfalse
    \let\@do@wrglossary\gls@noidxglossary
    \let\@gls@getgrouptitle\@gls@noidx@getgrouptitle
    \let\gls@checkseeallowed\relax
    \renewcommand{\@do@seeglossary}[2]{%
      \protected@edef\@gls@label{\glsdetoklabel{##1}}%
      \protected@write\@auxout{}{%
        \string\@gls@reference
        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}%
        {%
          \string\glsseeformat##2}%
        }%
      }%
    }%
  }%
  \AtBeginDocument
  {%
    \write\@auxout{\string\providecommand\string\@gls@reference[3]{}}%
  }%
  \def\warn@noprintglossary{%
    \GlossariesWarningNoLine{No \string\printnoidxglossary\space
      or \string\printnoidxglossaries ^^J
      found. (Remove \string\makenoidxglossaries\space if you
      don't want any glossaries.) ^^JThis document will not have a glossary}%
  }%
  \let\warn@nomakeglossaries\relax

```

```

\let\makeglossaries\@no@makeglossaries
\renewcommand*{\@printgloss@setsort}{%
  \let\@glo@assign@sortkey\@glo@assign@sortkey
  \def\@glo@sorttype{\@glo@default@sorttype}%
}%
\renewcommand*\new@glossaryentry[2]{%
  \PackageError{glossaries}{Glossary entries must be
    defined in the preamble^^Jwhen you use
    \string\makenoidxglossaries}%
  {Either move your definitions to the preamble or use
    \string\makeglossaries}%
}%
\renewcommand*{\glsentrynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \glsnoidxloclist{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}%
\renewcommand*{\glsdisplaynumberlist}[1]{%
  \letcs{\@gls@loclist}{glo@\glsdetoklabel{##1}@loclist}%
  \ifdef\@gls@loclist
  {%
    \def\@gls@noidxloclist@sep{%
      \def\@gls@noidxloclist@sep{%
        \def\@gls@noidxloclist@sep{%
          \glsnumlistsep
        }%
        \def\@gls@noidxloclist@finalsep{\glsnumlistlastsep}%
      }%
    }%
    \def\@gls@noidxloclist@finalsep{}%
    \def\@gls@noidxloclist@prev{}%
    \forlistloop{\glsnoidxdisplayloclisthandler}{\@gls@loclist}%
    \@gls@noidxloclist@finalsep
    \@gls@noidxloclist@prev
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
}
```

```

    }%
}%
\renewcommand*{\glsnumberlistloop}[3]{%
  \letcs{\@gls@loclist}{glo@glsdetoklabel{##1}@loclist}%
  \let\@gls@org@glsnoidxdisplayloc@glsnoidxdisplayloc
  \let\@gls@org@glsseeformat@glsseeformat
  \let@glsnoidxdisplayloc##2\relax
  \let@glsseeformat##3\relax
  \ifdef\@gls@loclist
  {%
    \forlistloop{\glsnoidxnumberlistloophandler}{\@gls@loclist}%
  }%
  {%
    ??\glsdoifexists{##1}%
    {%
      \GlossariesWarning{Missing location list for ‘##1’. Either
        a rerun is required or you haven’t referenced the entry}%
    }%
  }%
  \let@glsnoidxdisplayloc\@gls@org@glsnoidxdisplayloc
  \let@glsseeformat\@gls@org@glsseeformat
}%
\let\@gls@sanitizesort\@gls@noidx@sanitizesort
\let\@gls@nosanitizesort\@gls@noidx@nosanitizesort
\@gls@noidx@setsanitizesort
\@glo@check@sortallowed\makenoidxglossaries
\let\@gls@savenonumberlist\gls@noidx@nonumberlist
}%
}
\@onlypreamble{\makenoidxglossaries}
\newcommand*{\glsnumberlistloop}[3]{%
  \PackageError{glossaries}{\string\glsnumberlistloop\space
    only works with \string\makenoidxglossaries}{}%
}
\newcommand*{\glsnoidxnumberlistloophandler}[1]{%
  #1%
}
\newcommand*{\@no@makeglossaries}{%
  \PackageError{glossaries}{You can’t use both
    \string\makeglossaries\space and \string\makenoidxglossaries}%
  {Either use one or other (or none) of those commands but not both
    together.}%
}
\newcommand{\@gls@noref@warn}[1]{%
  \GlossariesWarning{\string\makenoidxglossaries\space
    is required to make \string\printnoidxglossary[type={#1}] work}%
}
\newcommand*{\gls@noidxglossary}{%
  \protected@write\@auxout{\gls@rglossdisablelocationcmds}{%
    \string\@gls@reference

```

```

        {\csname glo@\@gls@label @type\endcsname}%
        {\@gls@label}%
        {\string\glsnoidxdisplayloc
         {\@glo@counterprefix}%
         {\@gls@counter}%
         {\@glsnumberformat}%
         {\@glslocref}}%
      }%
    }%
  }
\AtEndDocument{%
  \glswritefiles
}
\newcommand*{\@glswritefiles}{%
  \forallglossaries{\@glo@type}{%
    \ifcsundef{glo@\@glo@type @filetok}%
    {%
      \def\gls@tmp{}%
    }%
    {%
      \edef\gls@tmp{\expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
    }%
    \ifx\gls@tmp\@empty
      \ifx\@glo@type\glsdefaulttype
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries.^^JRemember to use package option 'nomain' if
you
          don't want to^^Juse the main glossary}%
      \else
        \GlossariesWarningNoLine{Glossary '@@glo@type' has no
          entries}%
      \fi
    \else
      \@glsopenfile{\glswrite}{\@glo@type}%
      \immediate\write\glswrite{%
        \expandafter\the
        \csname glo@\@glo@type @filetok\endcsname}%
      \immediate\closeout\glswrite
    \fi
  }%
}
\newcommand*{\gls@glossary}[1]{%
  \@gls@glossary{#1}%
}
\newcommand*{\@gls@glossary}[2]{%
  \if@gls@debug
    \PackageInfo{glossaries}{wrglossary(#1)(#2)}%
  \fi
}

```



```

\newcommand{\@gls@renewglossary}{%
  \gdef\@gls@glossary##1{\@bsphack\begingroup\gls@wrglossary{##1}}%
  \let\@gls@renewglossary\@empty
}
\newcommand*{\gls@wrglossary}[2]{%
  \ifglssavewrites
    \protected@edef\@gls@tmp{\the\csname glo@#1@filetok\endcsname#2}%
    \expandafter\global\expandafter\csname glo@#1@filetok\endcsname
    \expandafter{\@gls@tmp~J}%
  \else
    \ifcsdef{glo@#1@file}%
    {%
      \expandafter\protected@write\csname glo@#1@file\endcsname{%
        \gls@disablepagerefexpansion}{#2}%
      }%
    {%
      \ifignoredglossary{#1}{}%
      {%
        \GlossariesWarning{No file defined for glossary ‘#1’}%
      }%
    }%
  \fi
  \endgroup\@esphack
}
\newcommand*{\@do@wrglossary}[1]{%
  \gls@ifnotmeasuring{\glswriteentry{#1}{\@do@wrglossary{#1}}}%
}
\newcommand*{\glswriteentry}[2]{%
  \ifglsindexonlyfirst
    \ifglsused{#1}{#2}%
  \else
    #2%
  \fi
}
\newcommand{\gls@protected@pagefmts}{\gls@numberpage,\gls@alphpage,%
\gls@Alphpage,\gls@romanpage,\gls@Romanpage,\gls@arabicpage}
\newcommand*{\gls@disablepagerefexpansion}{%
  \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\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{\theglscopycounter}%
  \expandafter\def\expandafter\@glsHlocref\expandafter{\theHglscopycounter}%
  \ifx\@glsHlocref\@glslocref
    \def\@glo@counterprefix{}%
  \else
    \@xp@gls@getcounterprefix{\@glslocref}{\@glsHlocref}%
  \fi
  \protected@edef\@gls@label{\glsdetoklabel{#1}}%

```

```

    \@@do@@wrglossary
}
\newif\ifglswrallowprimitivemods
\glswrallowprimitivemodsfalse
\newcommand*{\@@do@esc@wrglossary}[1]{% please read documented code!
\begingroup
  \let\gls@orgthe\the
  \let\gls@orgnumber\number
  \let\gls@orgarabic\@arabic
  \let\gls@orgromannumeral\romannumeral
  \let\gls@orgalph\@alph
  \let\gls@orgAlph\@Alph
  \let\gls@orgRoman\@Roman
  \ifglswrallowprimitivemods
    \def\gls@the##1{%
      \ifx##1\c@page \gls@numberpage\else\gls@orgthe##1\fi}%
    \def\the{\expandafter\gls@the}%
    \def\gls@number##1{%
      \ifx##1\c@page \gls@numberpage\else\gls@orgnumber##1\fi}%
    \def\number{\expandafter\gls@number}%
    \def\romannumeral##1{%
      \ifx##1\c@page \gls@romanpage \else \gls@orgromannumeral##1\fi}%
  \fi
  \def\@arabic##1{%
    \ifx##1\c@page \gls@arabicpage\else\gls@orgarabic##1\fi}%
  \def\@Roman##1{%
    \ifx##1\c@page \gls@Romanpage\else\gls@orgRoman##1\fi}%
  \def\@alph##1{%
    \ifx##1\c@page \gls@alphpage\else\gls@orgalph##1\fi}%
  \def\@Alph##1{%
    \ifx##1\c@page \gls@Alphpage\else\gls@orgAlph##1\fi}%
  \@wrglossarynumberhook
  \gls@disablepagerefexpansion
  \protected@xdef\@glslocref{\theglsentrycounter}%
\endgroup
\@gls@checkmkidxchars\@glslocref
\expandafter\ifx\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}{%
\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{\@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
\glswrglossdisableanchorcmds
\protected@edef\@do@gls@getcounterprefix{%
\noexpand\egroup
\noexpand\@gls@getcounterprefix{#1}{#2}%
}%
\@do@gls@getcounterprefix
}
\newcommand*\@gls@getcounterprefix[2]{%
\def\@gls@thisloc{#1}\def\@gls@thisHloc{#2}%
\ifx\@gls@thisloc\@gls@thisHloc
\def\@glo@counterprefix{}%
\else
\def\@gls@get@counterprefix##1.#1##2\end@getprefix{%
\def\@glo@tmp{##2}%
\ifx\@glo@tmp\@empty
\def\@glo@counterprefix{}%
\else
\def\@glo@counterprefix{##1}%
\fi
}%
\@gls@get@counterprefix#2.#1\end@getprefix
\ifx\@glo@counterprefix\@empty
\GlossariesWarning{Hyper target ‘#2’ can’t be formed by
prefixing^^Jlocation ‘#1’. You need to modify the
definition of \string\theH\@gls@counter^^Jotherwise you
will get the warning: “name{\@gls@counter.#1}’ has been^^J

```

```

        referenced but does not exist"%
    \fi
\fi
}
\newcommand{\@do@seeglossary}[2]{%
\def\@gls@xref{#2}%
\@onelevel@sanitize\@gls@xref
\@gls@checkmkidxchars\@gls@xref
\ifglsxindy
\gls@glossary{\csname glo@#1@type\endcsname}{%
(indexentry
:key (\csname glo@#1@index\endcsname)
:xref (\string"\@gls@xref\string")
:attr \string"see\string"
)
}%
\else
\gls@glossary{\csname glo@#1@type\endcsname}{%
\string\glossaryentry{\csname glo@#1@index\endcsname
\@gls@encapchar glsseeformat\@gls@xref}{Z}}%
\fi
}
\def\@gls@fixbraces#1#2#3\@nil{%
\ifx#2[\relax
\@gls@fixbraces#1#2#3\@end@fixbraces
\else
\def#1{#{#2#3}}%
\fi
}
\def\@gls@fixbraces#1[#2]#3\@end@fixbraces{%
\def#1{[#2]{#3}}%
}
\newrobustcmd*\@glssee}[3][\seename]{%
\@do@seeglossary{#2}{[#1]{#3}}
\newcommand*\@glssee}[3][\seename]{%
\glssee[#1]{#3}{#2}}
\newrobustcmd*\@glsseeformat}[3][\seename]{%
\emph{#1} \glsseelist{#2}}
\newrobustcmd*\@glsseelist}[1]{%
\let\@gls@dolast\relax
\let\@gls@donext\relax
\@for\@gls@thislabel:=#1\do{%
\ifx\@xfor@nextelement\@nnil
\@gls@dolast
\else
\@gls@donext
\fi
\expandafter\glsseeitem\expandafter{\@gls@thislabel}%
\let\@gls@dolast\glsseelastsep
\let\@gls@donext\glsseesep

```

```

}%
}
\newcommand*{\glsseelastsep}{\space\andname\space}
\newcommand*{\glsseesep}{, }
\newrobustcmd*{\glsseeitem}[1]{\glshyperlink[\glsseeitemformat{#1}]{#1}}
\newcommand*{\glsseeitemformat}[1]{%
  \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}{%
  \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{\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@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%
}%
\gls@printglossary@end
\global\let\glossaryentrynumbers\@org@glossaryentrynumbers
\global\let\warn@noprntglossary\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}%
\ifglxindy
\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}%
}%
}
\ExplSyntaxOn
\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
\cs_if_exist:NTF \datatool_sortwordseq:NN
{
\seq_new:N \l__glossaries_actual_sort_seq

```



```

\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 }
}
\seq_new:N \l__glossaries_sort_seq
\prop_new:N \l__glossaries_sort_prop
\msg_new:nnn { glossaries } { presort }
{
  Preparing ~ to ~ sort ~ glossary ~ ‘#1’ ~
  (total ~ number ~ of ~ referenced ~ entries: ~ #2 )
}
\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 }
}
\cs_new:Nn \__glossaries_add_sort_item:n
{
  \glstoifexists { #1 }
  {
    \seq_if_in:NnF \l__glossaries_sort_seq { #1 }
    {
      \ifglshasparent { #1 }
      {
        \exp_args:Nv \__glossaries_add_sort_item:n
        { glo@ #1 @parent }
        \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
        {
          \tl_clear:N \l__glossaries_sort_value_tl
        }
      }
    }
  }
}

```

```

        \tl_put_right:Nn \l__glossaries_sort_value_tl
        { \datatoolctrlboundary \datatoolasciistart }
    }
}
{
    \tl_clear:N \l__glossaries_sort_value_tl
}
\tl_put_right:Nv \l__glossaries_sort_value_tl
{ glo@ #1 @sort }
\prop_put:NnV \l__glossaries_sort_prop { #1 }
\l__glossaries_sort_value_tl
\seq_put_right:Nn \l__glossaries_sort_seq { #1 }
\seq_put_right:Ne \l__glossaries_actual_sort_seq
{
    \exp_not:N \__glossaries_actual:nn
    { \exp_not:V \l__glossaries_sort_value_tl }
    { #1 }
}
}
}
}
\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
}
\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
}
\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
}
\cs_new:Nn \__glossaries_sort_case:n
{
  \__glossaries_do_sort:Nn
    \__glossaries_sort_case_handler:nN
      { #1 }
}
\cs_new:Nn \__glossaries_sort_case_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \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:Nc #2 { \text_purify:n { #2 } \datatoolctrlboundary }
}
\cs_new:Nn \__glossaries_sort_nocase:n
{
  \__glossaries_do_sort:Nn
    \__glossaries_sort_nocase_handler:nN
      { #1 }
}
\cs_new:Nn \__glossaries_sort_nocase_handler:nN
{
  \__glossaries_fetch_sort:n { #1 }
  \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:Nc #2 { \text_purify:n { #2 } \datatoolctrlboundary }
}
\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:Nc \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:nn \__glossaries_actual_label:nn
\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
\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 }
}
}
}
\cs_new:Npn \@gls@fetch@lettergrp
{
\tl_if_head_eq_meaning:VNTF \@gls@sort \__datatool_sorted:nnn
{
\tl_set:Nx \@glo@thislettergrp
{ \exp_args:NV \DTLsortedletter \@gls@sort }
}
{
\exp_args:NVV \DTLassignlettergroup \@gls@sort \@gls@sort
\@glo@thislettergrp
}
}
}
{
\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
    }
  }
\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
\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{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{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}%
  {%
    \@glo@@sortentries{\glsorder}{#1}%
  }%
  {%
    \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{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{nocase}{#1}%
    }%
    {%
        \PackageError{glossaries}{Conflicting sort options:^^J
        \string\usepackage[sort=\@glo@default@sorttype]{glossaries}^^J
        \string\printnoidxglossary[sort=nocase]{}%
    }%
}
\newcommand*{\@glo@sortmacro@def}[1]{%
    \def\@glo@sortinglist{}%
    \forglseentries[#1]{\@gls@thislabel}%
    {%
        \xifinlistcs{\@gls@thislabel}{\@glsref@#1}%
        {%
            \listcsadd{\@glo@sortinglist}{\@gls@thislabel}%
        }%
        {%
        }%
    }%
    \cslet{\@glsref@#1}{\@glo@sortinglist}%
}
\newcommand*{\@glo@sortmacro@def@do}[1]{%
    \ifinlistcs{#1}{\@glsref@\@glo@type}%
    {}%
    {%
        \listcsadd{\@glsref@\@glo@type}{#1}%
    }%
    \ifcsdef{\@glo@sortingchildren@#1}%
    {%
        \@glo@addchildren{\@glo@type}{#1}%
    }%
    {}%
}
}

```



```

\newcommand*{\@glo@sortmacro@use}[1]{%
\ExplSyntaxOn
\newcommand*{\@print@noidx@glossary}{%
  \__glossaries_print_noidx:
}
\tl_new:N \l__glossaries_content_tl
\tl_new:N \l__glossaries_entry_prev_letter_group_tl
\cs_new:Nn \__glossaries_print_noidx:
{
  \tl_clear:N \l__glossaries_content_tl
  \cs_if_exist:cTF { @glsref@ \@glo@type }
  {
    \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
  \tl_put_right:Nn \l__glossaries_content_tl
  {
    \glossarysection [ \glossarytoctitle ] { \glossarytitle }
    \glossarypreamble
    \begin { theglossary }
    \glossaryheader
    \glsresetentrylist
  }
  \forlistcsloop \__glossaries_noidx_do:n { @glsref@ \@glo@type }
  \tl_put_right:Nn \l__glossaries_content_tl
  {
    \end { theglossary }
    \glossarypostamble
  }
  \glsnoidxprecontenthook
  \l__glossaries_content_tl
}
{
  \@gls@noref@warn { \@glo@type }
}
}
\newcommand\glsnoidxprecontenthook{}
\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 }
{
  \int_set:Nn \gls@level { \use:c { glo@\glsdetoklabel{#1}@level } }
  \tl_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:N \@gls@loclist }
        }
      }
    }
  }
}
}
{
  \tl_clear:N \l__glossaries_entry_letter_group_tl
  \tl_clear:N \l__glossaries_entry_letter_title_tl
  \tl_if_exist:cT
  { glo@ #1 @dtlsortgroup }
  {
    \tl_set:Ne \l__glossaries_entry_letter_group_tl
    {
      \exp_not:v { glo@ #1 @dtlsortgroup }
    }
    \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 \__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_group_title_tl }
      \l__glossaries_entry_letter_title_tl
    }
  }
}
\tl_if_empty:NT \l__glossaries_entry_letter_group_tl

```

```

{
  \letcs{\@gls@sort}{glo\glsdetoklabel{#1}@sort}%
  \@gls@fetch@lettergrp
  \tl_set_eq:NN
    \l__glossaries_entry_letter_group_tl
    \@glo@thislettergrp
  \tl_clear:c
    { \l__glossaries_noidx_#1_group_title_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
\newcommand*{\@gls@old@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
  }%
  {%
    \glstrifemptyglossary{\@glo@type}%
    {}%
    {\glossarysection[\glossarytoctitle]{\glossarytitle}}%
    \@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}%
\@gls@fetch@lettergrp
\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}%
}
\ExplSyntaxOn
\newcommand*{\@gls@reference}[3]{%
    \glsdoifexistsorwarn{#2}%
    {%
        \__glossaries_add_glsref:nnn { #1 } { #2 } { #3 }
    }%
}
\cs_new:Nn \__glossaries_add_glsref:nnn
{
    \__glossaries_add_glsref:nn { #1 } { #2 }
    \ifcsundef { glo@\glsdetoklabel{#2} @loclist }
    { \csgdef { glo@\glsdetoklabel{#2} @loclist } { } }
    {}%
}

```

```

\str_set:Nn \l__glossaries_current_location_str { #3 }
\cs_set_eq:NN \__glossaries_addloc:n \use:n
\forlistcsloop \__glossaries_check_loc_do:n
{ glo@ \glsdetoklabel{#2} @loclist }
\__glossaries_addloc:n
{ \listcsgadd { glo@ \glsdetoklabel{#2}@ loclist } { #3 } }
}
\str_new:N \l__glossaries_current_location_str
\cs_new:Nn \__glossaries_check_loc_do:n
{
\str_if_eq:VnT \l__glossaries_current_location_str { #1 }
{
\cs_set_eq:NN \__glossaries_addloc:n \use_none:n
\listbreak
}
}
\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
\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}{}%
    \renewcommand*{\@glossaryseclabel}{}%
  \or
    \renewcommand*{\@glossarysecstar}{}%
    \renewcommand*{\@glossaryseclabel}{\label{\glsautoprefix\@glo@type}}%
  \or
    \renewcommand*{\@glossarysecstar}{*}%
    \renewcommand*{\@glossaryseclabel}{%
      \protected@edef\@currentlabelname{\glossarytoctitle}%
      \label{\glsautoprefix\@glo@type}}%
  \fi
}
\define@choicekey{printgloss}{nogroupskip}{true,false}[true]{%
  \csuse{glsnogroupskip#1}%
}
\define@choicekey{printgloss}{nopostdot}{true,false}[true]{%
  \csuse{glsnopostdot#1}%
}
\newcommand*{\GlsEntryCounterLabelPrefix}{glsentry-}
\define@choicekey{printgloss}{entrycounter}{true,false}[true]{%
  \csuse{glsentrycounter#1}%
  \@gls@define@glossaryentrycounter
}
\define@choicekey{printgloss}{subentrycounter}{true,false}[true]{%
  \csuse{glssubentrycounter#1}%
  \@gls@define@glossarysubentrycounter
}
\define@boolkey{printgloss}[gls]{nonumberlist}[true]{%
\ifglsnonumberlist
  \def\glossaryentrynumbers##1{%
\else
  \def\glossaryentrynumbers##1{##1}%
\fi}

```



```

\define@key{printgloss}{sort}{\@glo@assign@sortkey{#1}}
\newcommand*{\@glo@no@assign@sortkey}[1]{%
  \PackageError{glossaries}{‘sort’ key not permitted with
  \string\printglossary}%
  {The ‘sort’ key may only be used with \string\printnoidxglossary}%
}
\newcommand*{\@glo@assign@sortkey}[1]{%
  \def\@glo@sorttype{#1}%
}
\newcommand*{\@glsnonextpages}{%
  \gdef\glossaryentrynumbers##1{%
    \glsresetentrylist
  }%
}
\newcommand*{\@glsnextpages}{%
  \gdef\glossaryentrynumbers##1{%
    ##1\glsresetentrylist}}
\newcommand*{\glsresetentrylist}{%
  \global\let\glossaryentrynumbers\org@glossaryentrynumbers}
\newcommand*{\glsnonextpages}{}
\newcommand*{\glsnextpages}{}
\@gls@define@glossaryentrycounter
\@gls@define@glossarysubentrycounter
\newcommand*{\glsresetsubentrycounter}{%
  \ifglssubentrycounter
    \setcounter{glossarysubentry}{0}%
  \fi
}
\newcommand*{\glsresetentrycounter}{%
  \ifglsentrycounter
    \setcounter{glossaryentry}{0}%
  \fi
}
\newcommand*{\glsstepentry}[1]{%
  \ifglsentrycounter
    \refstepcounter{glossaryentry}%
    \label{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \fi
}
\newcommand*{\glsstepsubentry}[1]{%
  \ifglssubentrycounter
    \protected@edef\currentglssubentry{\glsdetoklabel{#1}}%
    \refstepcounter{glossarysubentry}%
    \label{\GlsEntryCounterLabelPrefix\currentglssubentry}%
  \fi
}
\newcommand*{\glsrefentry}[1]{%
  \ifglsentrycounter
    \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
  \else

```

```

\ifglssubentrycounter
  \ref{\GlsEntryCounterLabelPrefix\glsdetoklabel{#1}}%
\else
  \gls{#1}%
\fi
\fi
}
\newcommand*{\glentrycounterlabel}{%
  \ifglentrycounter
    \theglossaryentry.\space
  \fi
}
\newcommand*{\glssubentrycounterlabel}{%
  \ifglssubentrycounter
    \theglossarysubentry)\space
  \fi
}
\newcommand*{\glentryitem}[1]{%
  \ifglentrycounter
    \glstepentry{#1}\glentrycounterlabel
  \else
    \glsresetsubentrycounter
  \fi
}
\newcommand*{\glssubentryitem}[1]{%
  \ifglssubentrycounter
    \glstepsubentry{#1}\glssubentrycounterlabel
  \fi
}
\ifcsundef{theglossary}%
{%
  \newenvironment{theglossary}{}{}%
}%
{%
  \@gls@warnontheGLOSSdefined
  \renewenvironment{theglossary}{}{}%
}
\newcommand*{\glossaryheader}{}
\newcommand*{\glstarget}[2]{\@glstarget{\glolinkprefix#1}{#2}}
\newcommand{\glossentry}[2]{\glossentryname{#1} #2}
\newcommand*{\glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%
    \letcs{\glo@name}{\glo@\glsdetoklabel{#1}@name}%
    \expandafter\glsnamefont\expandafter{\glo@name}%
  }%
}
\newcommand*{\Glossentryname}[1]{%
  \glsdoifexistsorwarn{#1}%
  {%

```

```

\glsnamefont{\Glsentryname{#1}}%
}%
}
\glsmfuaddmap{\glossentryname}{\Glossentryname}
\newcommand*{\glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsentrydesc{#1}%
}%
}
\newcommand*{\Glossentrydesc}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\Glsentrydesc{#1}%
}%
}
\glsmfuaddmap{\glossentrydesc}{\Glossentrydesc}
\newcommand*{\glossentrysymbol}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\glsentrysymbol{#1}%
}%
}
\newcommand*{\Glossentrysymbol}[1]{%
\glsdoifexistsorwarn{#1}%
{%
\Glsentrysymbol{#1}%
}%
}
\glsmfuaddmap{\glossentrysymbol}{\Glossentrysymbol}
\newcommand{\subglossentry}[3]{\glossentryname{#2} #3}
\newcommand*{\glsgroupskip}{}
\newcommand*{\glsgroupheading}[1]{}
\newrobustcmd*{\glsgetgrouptitle}[1]{%
\@gls@getgrouptitle{#1}{\@gls@grptitle}%
\@gls@grptitle
}
\ExplSyntaxOn
\newcommand*{\@gls@getgrouptitle}[2]{%
\def#2{#1}%
\regex_match:nnT { ~\A(.|glsnumbers|glssymbols)\Z } { #1 }
{ \ifcsundef{#1groupname}{ }\letcs#2{#1groupname} } }
}
\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
\ExplSyntaxOn
\newcommand*{\glsgetgrouplabel}[1]{%
  \str_if_eq:eeTF {#1} { \glssymbolsgroupname }
  { glssymbols }
  {
    \str_if_eq:eeTF {#1} { \glsnumbersgroupname }
    { glsnumbers }
    { #1 }
  }
}
\ExplSyntaxOff
\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{\newglossarystyle}[2]{%
  \ifcsundef{@glsstyle@#1}%
  {%
    \expandafter\def\csname @glsstyle@#1\endcsname{#2}%
  }%
  {%
    \PackageError{glossaries}{Glossary style ‘#1’ is already defined}{}%
  }%
}
\newcommand{\renewglossarystyle}[2]{%

```

```

\ifcsundef{@glsstyle@#1}%
{%
  \PackageError{glossaries}{Glossary style ‘#1’ isn’t already defined}{}%
}%
{%
  \csdef{@glsstyle@#1}{#2}%
}%
}
\newcommand*{\glsnamefont}[1]{#1}
\ifcsundef{hyperlink}%
{%
  \def\glshypernumber#1{#1}%
}%
{%
  \def\glshypernumber#1{\xp@glshypernumber{#1}}
}
\newrobustcmd{\xp@glshypernumber}[1]{%
  \@glshypernumber#1\nohyperpage{}\@nil
}
\def\@glshypernumber#1\nohyperpage#2#3\@nil{%
  \ifx\\#1\\%
    \else
      \gls@delimR#1\delimR\delimR\\%
    \fi
    \ifx\\#2\\%
      \else
        #2%
      \fi
      \ifx\\#3\\%
        \else
          \@glshypernumber#3\@nil
        \fi
      \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{\glshentrycounter\@glo@counterprefix
\glswrglosslocationtarget{\@gls@x}}%
\@onelevel@sanitize\@gls@tmp
\edef\@gls@x{\noexpand\egroup
\noexpand\hyperlink{\@gls@tmp}%
{noexpand\glswrglosslocationtextfmt{\the\toks@}}}%
}%
\@gls@x
\fi
\else
\@gls@ReturnAfterFi{%
\@gls@removespaces#2\@nil
}%
\fi
}
\long\def\@gls@ReturnAfterFi#1\fi{\fi#1}
\newcommand*\hyperrm[1]{\textrm{\glshypernumber{#1}}}
\newcommand*\hypersf[1]{\textsf{\glshypernumber{#1}}}
\newcommand*\hypertt[1]{\texttt{\glshypernumber{#1}}}
\newcommand*\hyperbf[1]{\textbf{\glshypernumber{#1}}}
\newcommand*\hypermd[1]{\textmd{\glshypernumber{#1}}}
\newcommand*\hyperit[1]{\textit{\glshypernumber{#1}}}
\newcommand*\hypersl[1]{\textsl{\glshypernumber{#1}}}
\newcommand*\hyperup[1]{\textup{\glshypernumber{#1}}}
\newcommand*\hypersc[1]{\textsc{\glshypernumber{#1}}}
\newcommand*\hyperemph[1]{\emph{\glshypernumber{#1}}}
\newcommand{\oldacronym}[4][\@gls@label]{%
\def\@gls@label{#2}%
\newacronym[#4]{#1}{#2}{#3}%
\ifcsundef{xspace}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{#1}}}%
}%
}%
{%
\expandafter\edef\csname#1\endcsname{%
\noexpand\@ifstar{\noexpand\Gls{#1}}{\noexpand\gls{xspace}}}%
}

```

```

        \noexpand\gls{#1}\noexpand\xspace}%
    }%
}%
}
\newcommand{\newacronym}[4][{}]{%
  \gls@newacronym{#1}{#2}{#3}{#4}%
}
\newcommand{\gls@newacronym}[4]{%
  \renewcommand{\gls@newacronym}[4]{%
    \PackageError{glossaries}{incompatible acronym style:
      rollback may be required}%
    {something has gone wrong with \string\newacronym. Try
      \string\usepackage{glossaries}[=v4.49]}%
  }%
  \setacronymstyle{long-short}%
  \newacronym[#1]{#2}{#3}{#4}%
}
\newcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}
\newrobustcmd*{\glstextup}[1]{\ifdef\textulc{\textulc{#1}}{\textup{#1}}}
\newcommand*{\glsshortkey}{short}
\newcommand*{\glsshortpluralkey}{shortplural}
\newcommand*{\glslongkey}{long}
\newcommand*{\glslongpluralkey}{longplural}
\newcommand{\acrlinkfullformat}[5]{%
  \acrfullformat{#1}{#3}{#4}{#5}{#2}{#3}{#4}[]}%
}
\newcommand{\acrfullformat}[2]{#1\glsspace{#2}}
\newrobustcmd{\glsspace}{\space}
\newcommand*{\acrfullfmt}[3]{%
  \acrlinkfullformat{\@acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullfmt}[3]{%
  \acrlinkfullformat{\@Acrlong}{\@acrshort}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullfmt}[3]{%
  \acrlinkfullformat{\@ACRlong}{\@ACRshort}{#1}{#2}{#3}%
}
\newcommand*{\acrfullplfmt}[3]{%
  \acrlinkfullformat{\@acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\Acrfullplfmt}[3]{%
  \acrlinkfullformat{\@Acrlongpl}{\@acrshortpl}{#1}{#2}{#3}%
}
\newcommand*{\ACRfullplfmt}[3]{%
  \acrlinkfullformat{\@ACRlongpl}{\@ACRshortpl}{#1}{#2}{#3}%
}
\newrobustcmd*{\acrfull}{\@gls@hyp@opt\ns@acrfull}
\newcommand*{\ns@acrfull}[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}{%
    {\@acrfull{#1}{#2}[]}%
  }
}

```

```

}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfull{\@gls@hyp@opt\@ns@acrfull}
\glsmfuaddmap{\acrfull}{\@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfull{\@gls@hyp@opt\@ns@acrfull}
\newcommand*\ns@acrfull[2][{}]{%
  \new@ifnextchar[{\@acrfull{#1}{#2}}%
    {\@acrfull{#1}{#2}[]}%
}
\def\@acrfull#1#2[#3]{%
  \acrfullfmt{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\@ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\@ns@acrfullpl}
\glsmfuaddmap{\acrfullpl}{\@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newrobustcmd*\@acrfullpl{\@gls@hyp@opt\@ns@acrfullpl}
\newcommand*\ns@acrfullpl[2][{}]{%
  \new@ifnextchar[{\@acrfullpl{#1}{#2}}%
    {\@acrfullpl{#1}{#2}[]}%
}
\def\@acrfullpl#1#2[#3]{%
  \acrfullplfmt{#1}{#2}{#3}%
}
\newcommand{\acronymfont}[1]{#1}
\newcommand{\firstacronymfont}[1]{\acronymfont{#1}}
\newcommand*\acronymformat[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={\acronymssort{\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}{%
    \glssupercase{\genacrfullformat{##2}{##3}}}%
}%
\renewcommand*{\acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\genplacrfullformat{##2}{##3}}}%
\renewcommand*{\Acrfullplfmt}[3]{%
  \glslink[##1]{##2}{\Genplacrfullformat{##2}{##3}}}%
\renewcommand*{\ACRfullplfmt}[3]{%
  \glslink[##1]{##2}{%
    \glssupercase{\genplacrfullformat{##2}{##3}}}%
}%

```

```

\renewcommand*{\glsentryfull}[1]{\genacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfull}[1]{\Genacrfullformat{##1}{}}%
\renewcommand*{\glsentryfullpl}[1]{\genplacrfullformat{##1}{}}%
\renewcommand*{\Glsentryfullpl}[1]{\Genplacrfullformat{##1}{}}%
}
\newcommand*{\GenericAcronymFields}{description={\the\glslongtok}}
\newcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{#1}}}
\newcommand*{\acronymsort}[2]{#1}
\newcommand*{\setacronymstyle}[1]{%
  \ifcsundef{@glsacr@dispstyle@#1}
  {%
    \PackageError{glossaries}{Undefined acronym style ‘#1’}{}%
  }%
  {%
    \ifdefempty{\@glsacronymlists}%
    {%
      \DeclareAcronymList{\acronymtype}%
    }%
    {}%
    \SetGenericNewAcronym
    \GlsUseAcrStyleDefs{#1}%
    \@for\@gls@type:=\@glsacronymlists\do{%
      \defglsentryfmt[\@gls@type]{\GlsUseAcrEntryDispStyle{#1}}%
    }%
  }%
}
\newcommand*{\newacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ already exists}{}%
  }%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
}
\newcommand*{\renewacronymstyle}[3]{%
  \ifcsdef{@glsacr@dispstyle@#1}%
  {%
    \csdef{@glsacr@dispstyle@#1}{#2}%
    \csdef{@glsacr@styledefs@#1}{#3}%
  }%
  {%
    \PackageError{glossaries}{Acronym style ‘#1’ doesn’t exist}{}%
  }%
}
\newcommand*{\GlsUseAcrEntryDispStyle}[1]{\csuse{@glsacr@dispstyle@#1}}
\newcommand*{\GlsUseAcrStyleDefs}[1]{\csuse{@glsacr@styledefs@#1}}
\newacronymstyle{long-short}%
%
```

```

\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glstrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glstrylong{##1}##2\space
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glstrylongpl{##1}##2\space
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glstryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sp-short}%
{%
\ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\glstrylong{##1}##2\glssacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\Glstrylong{##1}##2\glssacspace{##1}%
(\protect\firstacronymfont{\glstryshort{##1}})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\glstrylongpl{##1}##2\glssacspace{##1}%
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\Glstrylongpl{##1}##2\glssacspace{##1}%
(\protect\firstacronymfont{\glstryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glstryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%

```

```

\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newcommand*{\glsacspace}[1]{%
\settowidth{\dimen@}{(\firstacronymfont{\glsentryshort{##1}})}%
\ifdim\dimen@<3em~\else\space\fi
}
\newacronymstyle{short-long}%
{%
\ifglsahaslong{\glslabel}{\glsгенacfmt}{\glsгенentryfmt}%
}%
{%
\renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
\renewcommand*{\genacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*{\Genacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshort{##1}}##2\space
(\glsentrylong{##1})%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2\space
(\glsentrylongpl{##1})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\firstacronymfont}[1]{\acronymfont{##1}}%
\renewcommand*{\acrpluralsuffix}{\glspluralsuffix}%
}
\newacronymstyle{long-sc-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{long-sm-short}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%

```

```

\GlsUseAcrStyleDefs{long-short}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{sc-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}
\newacronymstyle{sm-short-long}%
{%
\GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
\GlsUseAcrStyleDefs{short-long}%
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{long-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-short}%
}%
{%
\GlsUseAcrStyleDefs{long-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\space (\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sp-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sp-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sp-short}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glsentrylong{##1}\glsacspace{##1}(\acronymfont{\glsentryshort{##1}})}%
}
\newacronymstyle{long-sc-short-desc}%
{%
\GlsUseAcrEntryDisplayStyle{long-sc-short}%
}%
{%
\GlsUseAcrStyleDefs{long-sc-short}%
}

```

```

\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
  \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{long-sm-short-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{long-sm-short}%
}%
{%
  \GlsUseAcrStyleDefs{long-sm-short}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{short-long}%
}%
{%
  \GlsUseAcrStyleDefs{short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sc-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sc-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sc-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{sm-short-long-desc}%
{%
  \GlsUseAcrEntryDisplayStyle{sm-short-long}%
}%
{%
  \GlsUseAcrStyleDefs{sm-short-long}%
  \renewcommand*{\GenericAcronymFields}{}%
  \renewcommand*{\acronymsort}[2]{##2}%
  \renewcommand*{\acronymentry}[1]{%
    \glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
  }
\newacronymstyle{dua}%

```

```

{%
  \ifdefempty\glscustomtext
  {%
    \ifglshaslong{\glslabel}%
    {%
      \glsifplural
      {%
        \glscapscase
        {%
          \glstrylongpl{\glslabel}\glsinsert
        }%
        {%
          \Glsentrylongpl{\glslabel}\glsinsert
        }%
        {%
          \glssupercase
          {\glstrylongpl{\glslabel}\glsinsert}%
        }%
      }%
    }%
    {%
      \glscapscase
      {%
        \glstrylong{\glslabel}\glsinsert
      }%
      {%
        \Glsentrylong{\glslabel}\glsinsert
      }%
      {%
        \glssupercase
        {\glstrylong{\glslabel}\glsinsert}%
      }%
    }%
  }%
  {%
    \glsgenentryfmt
  }%
  {\glscustomtext\glsinsert}%
}%
{%
  \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
  \renewcommand*{\acrfullfmt}[3]{%
    \glslink[##1]{##2}{\glstrylong{##2}##3\space
      (\acronymfont{\glstryshort{##2}})}%
  }%
  \renewcommand*{\Acrfullfmt}[3]{%
    \glslink[##1]{##2}{\Glsentrylong{##2}##3\space
      (\acronymfont{\glstryshort{##2}})}%
  }%
  \renewcommand*{\ACRfullfmt}[3]{%
    \glslink[##1]{##2}{%
      \glssupercase{\glstrylong{##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}{%
        \glssupercase{\glsentrylongpl{##2}##3\space
            (\acronymfont{\glsentryshortpl{##2}})}}}%
\renewcommand*{\glsentryfull}[1]{%
    \glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\Glsentryfull}[1]{%
    \Glsentrylong{##1}\space(\acronymfont{\glsentryshort{##1}})%
}%
\renewcommand*{\glsentryfullpl}[1]{%
    \glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\Glsentryfullpl}[1]{%
    \Glsentrylongpl{##1}\space(\acronymfont{\glsentryshortpl{##1}})%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glsentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glsacrpluralsuffix}%
}
\newacronymstyle{dua-desc}%
{%
    \GlsUseAcrEntryDispStyle{dua}%
}%
{%
    \GlsUseAcrStyleDefs{dua}%
    \renewcommand*{\GenericAcronymFields}{}%
    \renewcommand*{\acronymentry}[1]{\glsentrylong{##1}}%
    \renewcommand*{\acronymsort}[2]{##2}%
}%
\newacronymstyle{footnote}%
{%
    \ifglshaslong{\glslabel}{\glsgenacfmt}{\glsgenentryfmt}%
}%
{%
    \renewcommand*{\GenericAcronymFields}{description={\the\glslongtok}}%
    \glshyperfirstfalse
    \renewcommand*{\genacrfullformat}[2]{%
        \protect\firstacronymfont{\glsentryshort{##1}}##2%
        \protect\footnote{\glsentrylong{##1}}%
    }%
    \renewcommand*{\Genacrfullformat}[2]{%

```



```

\firstacronymfont{\Glsentryshort{##1}}##2%
\protect\footnote{\glentrylong{##1}}%
}%
\renewcommand*{\genplacrfullformat}[2]{%
\protect\firstacronymfont{\glentryshortpl{##1}}##2%
\protect\footnote{\glentrylongpl{##1}}%
}%
\renewcommand*{\Genplacrfullformat}[2]{%
\protect\firstacronymfont{\Glsentryshortpl{##1}}##2%
\protect\footnote{\glentrylongpl{##1}}%
}%
\renewcommand*{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}%
\renewcommand*{\acronymsort}[2]{##1}%
\renewcommand*{\acronymfont}[1]{##1}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
\renewcommand*{\acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glentryshort{##2}}##3\space
(\glentrylong{##2})}%
\renewcommand*{\Acrfullfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshort{##2}}##3\space
(\glentrylong{##2})}%
\renewcommand*{\ACRfullfmt}[3]{%
\glslink[##1]{##2}{%
\glssupercase{\acronymfont{\glentryshort{##2}}##3\space
(\glentrylong{##2})}}}%
\renewcommand*{\acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\glentryshortpl{##2}}##3\space
(\glentrylongpl{##2})}%
\renewcommand*{\Acrfullplfmt}[3]{%
\glslink[##1]{##2}{\acronymfont{\Glsentryshortpl{##2}}##3\space
(\glentrylongpl{##2})}%
\renewcommand*{\ACRfullplfmt}[3]{%
\glslink[##1]{##2}{%
\glssupercase{\acronymfont{\glentryshortpl{##2}}##3\space
(\glentrylongpl{##2})}}}%
\renewcommand*{\glentryfull}[1]{%
\acronymfont{\glentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*{\Glsentryfull}[1]{%
\acronymfont{\Glsentryshort{##1}}\space(\glentrylong{##1})}%
\renewcommand*{\glentryfullpl}[1]{%
\acronymfont{\glentryshortpl{##1}}\space(\glentrylongpl{##1})}%
\renewcommand*{\Glsentryfullpl}[1]{%
\acronymfont{\Glsentryshortpl{##1}}\space(\glentrylongpl{##1})}%
}
\newacronymstyle{footnote-sc}%
{%
\GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
}

```

```

\renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
\renewcommand{\acronymfont}[1]{\textsc{##1}}%
\renewcommand*{\acrpluralsuffix}{\glsupacrpluralsuffix}%
}%
\newacronymstyle{footnote-sm}%
{%
\GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand{\acronymentry}[1]{\acronymfont{\glentryshort{##1}}}
\renewcommand{\acronymfont}[1]{\textsmaller{##1}}%
\renewcommand*{\acrpluralsuffix}{\glacrpluralsuffix}%
}%
\newacronymstyle{footnote-desc}%
{%
\GlsUseAcrEntryDisplayStyle{footnote}%
}%
{%
\GlsUseAcrStyleDefs{footnote}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sc-desc}%
{%
\GlsUseAcrEntryDisplayStyle{footnote-sc}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sc}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newacronymstyle{footnote-sm-desc}%
{%
\GlsUseAcrEntryDisplayStyle{footnote-sm}%
}%
{%
\GlsUseAcrStyleDefs{footnote-sm}%
\renewcommand*{\GenericAcronymFields}{}%
\renewcommand*{\acronymsort}[2]{##2}%
\renewcommand*{\acronymentry}[1]{%
\glentrylong{##1}\space (\acronymfont{\glentryshort{##1}})}%
}
\newcommand*{\DefineAcronymSynonyms}{%
\let\acs\acrshort
\let\Acs\Acrshort

```

```

\let\acsp\acrshortpl
\let\Acsp\Acrshortpl
\let\acl\acrlong
\let\aclp\acrlongpl
\let\Acl\Acrlong
\let\Aclp\Acrlongpl
\let\acf\acrfull
\let\acfp\acrfullpl
\let\Acf\Acrfull
\let\Acfp\Acrfullpl
\let\ac\gls
\let\Ac\Gls
\let\acp\glspl
\let\Acp\Glspl
}
\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*{\acrnofootnote}[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}
\@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
}

```

9.3 Rollback v4.58 (glossary-mcols-2025-03-19.sty)

```

\NeedsTeXFormat{LaTeX2e}
\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{v4.58}{2025-03-19}
\ProvidesPackage{glossary-mcols}[2025/03/19 v4.58 (NLCT)]
\RequirePackage{multicol}
\RequirePackage{glossary-tree}[=v4.58]
\providecommand{\indexspace}{%
  \par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glsmcols}{2}
\newglossarystyle{mcolindex}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}
      \setlength{\parindent}{0pt}%
      \setlength{\parskip}{0pt plus 0.3pt}%
      \let\item\glstreeitem
      \let\subitem\glstreesubitem
      \let\subsubitem\glstreesubsubitem
    }%
    {\end{multicols}}%
}
\newglossarystyle{mcolindexgroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}\indexspace}%
}
\newglossarystyle{mcolindexhypergroup}{%
  \setglossarystyle{mcolindex}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\glsnavigationhypertarget{##1}{\glsgrouptitle{##1}}}%
    \indexspace}%
}
\newglossarystyle{mcolindexspannav}{%
  \setglossarystyle{index}%
  \renewenvironment{theglossary}%
    {%
      \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
      \setlength{\parindent}{0pt}%

```

```

\setlength{\parskip}{0pt plus 0.3pt}%
\let\item\glstreeitem}%
{\end{multicols}}}%
\renewcommand*{\glsgroupheading}[1]{%
\item\glstreegroupheaderfmt
{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}%
\indexspace}%
}
\newglossarystyle{mcoltree}{%
\setglossarystyle{tree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}
\setlength{\parindent}{0pt}%
\setlength{\parskip}{0pt plus 0.3pt}%
}%
{\end{multicols}}}%
}
\newglossarystyle{mcoltreegroup}{%
\setglossarystyle{mcoltree}%
\renewcommand{\glsgroupheading}[1]{\par
\noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreehypergroup}{%
\setglossarystyle{mcoltree}%
\renewcommand*{\glossaryheader}{%
\par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par\noindent
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##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}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcoltreenoname}{%
\setglossarystyle{treenoname}%
\renewenvironment{theglossary}%
{%

```

```

        \begin{multicols}{\glsmcols}
        \setlength{\parindent}{0pt}%
        \setlength{\parskip}{0pt plus 0.3pt}%
    }%
    {\end{multicols}}%
}
\newglossarystyle{mcoltreenonamegroup}{%
    \setglossarystyle{mcoltreenoname}%
    \renewcommand{\glsgroupeheading}[1]{\par
        \noindent\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcoltreenonamehypergroup}{%
    \setglossarystyle{mcoltreenoname}%
    \renewcommand*{\glossaryheader}{%
        \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    \renewcommand*{\glsgroupeheading}[1]{%
        \par\noindent
        \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
        \indexspace}%
}
\newglossarystyle{mcoltreenonamespannav}{%
    \setglossarystyle{treenoname}%
    \renewenvironment{theglossary}{%
        {%
            \begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
            \setlength{\parindent}{0pt}%
            \setlength{\parskip}{0pt plus 0.3pt}%
        }%
        {\end{multicols}}%
        \renewcommand*{\glsgroupeheading}[1]{%
            \par\noindent
            \glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
            \indexspace}%
    }
}
\newglossarystyle{mcolalttree}{%
    \setglossarystyle{alttree}%
    \renewenvironment{theglossary}{%
        {%
            \begin{multicols}{\glsmcols}
            \def\@gls@prevlevel{-1}%
            \mbox{}\par
        }%
        {\par\end{multicols}}%
    }
}
\newglossarystyle{mcolalttreegroup}{%
    \setglossarystyle{mcolalttree}%
    \renewcommand{\glsgroupeheading}[1]{\par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
    }
}

```



```

\glstreegroupheaderfmt{\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{mcolalttreehypergroup}{%
\setglossarystyle{mcolalttree}%
\renewcommand*{\glossaryheader}{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}
\newglossarystyle{mcolalttreespannav}{%
\setglossarystyle{alttree}%
\renewenvironment{theglossary}%
{%
\begin{multicols}{\glsmcols}[\noindent\glstreenavigationfmt{\glsnavigation}]
\def\@gls@prevlevel{-1}%
\mbox{}\par
}%
{\par\end{multicols}}%
\renewcommand*{\glsgroupheading}[1]{%
\par
\def\@gls@prevlevel{-1}%
\hangindent0pt\relax
\parindent0pt\relax
\glstreegroupheaderfmt{\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
\indexspace}%
}

```

9.4 Rollback v4.58 (glossary-tree-2025-03-19.sty)

```

\NeedsTeXFormat{LaTeX2e}
\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{v4.58}{2025-03-19}
\ProvidesPackage{glossary-tree}[2025/03/19 v4.58 (NLCT)]
\providecommand{\indexspace}{%
\par \vskip 10\p@ \@plus 5\p@ \@minus 3\p@ \relax
}
\newcommand*{\glstreenamefmt}[1]{\textbf{#1}}
\newcommand*{\glstreegroupheaderfmt}[1]{\glstreenamefmt{#1}}
\newcommand*{\glstreenavigationfmt}[1]{\glstreenamefmt{#1}}
\ifdef\@idxitem

```

```

{\newcommand{\glstreeitem}{\@idxitem}}
{\newcommand{\glstreeitem}{\par\hangindent40\p@}}
\ifdef\subitem
{\let\glstreesubitem\subitem}
{\newcommand\glstreesubitem{\glstreeitem\hspace*{20\p@}}}
\ifdef\subsubitem
{\let\glstreesubsubitem\subsubitem}
{\newcommand\glstreesubsubitem{\glstreeitem\hspace*{30\p@}}}
\newcommand{\glstreepredesc}{\space}
\newcommand{\glstreechildpredesc}{\space}
\newglossarystyle{index}{%
  \renewenvironment{theglossary}%
    {\setlength{\parindent}{0pt}%
     \setlength{\parskip}{0pt plus 0.3pt}%
     \let\item\glstreeitem
     \let\subitem\glstreesubitem
     \let\subsubitem\glstreesubsubitem
    }%
    {\par}%
  \renewcommand*{\glossaryheader}{}%
  \renewcommand*{\glsgroupheading}[1]{}%
  \renewcommand*{\glossentry}[2]{%
    \item\glsentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space(\glossentrysymbol{##1})}{}%
    \glstreepredesc \glossentrydesc{##1}\glspostdescription\space ##2%
  }%
  \renewcommand{\subglossentry}[3]{%
    \ifcase##1\relax
      % level 0
      \item
    \or
      % level 1
      \subitem
      \glssubentryitem{##2}%
    \else
      % all other levels
      \subsubitem
    \fi
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}%
    \ifglshassymbol{##2}{\space(\glossentrysymbol{##2})}{}%
    \glstreechildpredesc\glossentrydesc{##2}\glspostdescription\space ##3%
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}}
\newglossarystyle{indexgroup}{%
  \setglossarystyle{index}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt{\glsgrouptitle{##1}}%
    \indexspace
  }%
}

```

```

\newglossarystyle{indexhypergroup}{%
  \setglossarystyle{index}%
  \renewcommand*{\glossaryheader}{%
    \item\glstreenavigationfmt{\glsnavigation}\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \item\glstreegroupheaderfmt
      {\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
    \glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}%
    \ifglshassymbol{##1}{\space\glossentrysymbol{##1}}{}%
    \glstreepredesc\glossentrydesc{##1}\glspostdescription\space##2\par
  }%
  \renewcommand{\subglossentry}[3]{%
    \hangindent##1\glstreeindent\relax
    \parindent##1\glstreeindent\relax
    \ifnum##1=1\relax
      \glssubentryitem{##2}%
    \fi
    \glstarget{##2}{\strut}%
    \glossentrydesc{##2}\glspostdescription\space##3\par
  }%
  \renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{treenonamegroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand{\glsgroupheading}[1]{\par
    \noindent\glstreegroupheaderfmt
    {\glsgrouptitle{##1}}\par\indexspace}%
}
\newglossarystyle{treenonamehypergroup}{%
  \setglossarystyle{treenoname}%
  \renewcommand*{\glossaryheader}{%
    \par\noindent\glstreenavigationfmt{\glsnavigation}\par\indexspace}%
  \renewcommand*{\glsgroupheading}[1]{%
    \par\noindent
    \glstreegroupheaderfmt
    {\glsnavhypertarget{##1}{\glsgrouptitle{##1}}}\par
    \indexspace}%
}
\newrobustcmd*{\glsfindwidesttoplevelname}[1][\@glo@types]{%
  \dimen@=0pt\relax
  \glstmplen=0pt\relax
  \forall glossaries[#1]{\@gls@type}%
  {%
    \forall glsentries[\@gls@type]{\@glo@label}%
    {%

```

```

\ifglshasparent{\@glo@label}%
}%
{%
\glsmasurewidth{\dimen@}%
{\glstreenamefmt{\glstentryname{\@glo@label}}}%
\ifdim\dimen@>\gls@tmplen
\gls@tmplen=\dimen@
\letcs{\@glswidestname}{glo@\glsdetoklabel{\@glo@label}@name}%
\fi
}%
}%
}%
}
\newcommand*{\glssetwidest}[2][0]{%
\expandafter\def\csname @glswidestname\romannumeral#1\endcsname{%
#2}%
}
\newcommand*{\@glswidestname}{}
\newcommand*{\glstreenamebox}[2]{%
\makebox[#1][l]{#2}%
}
\newglossarystyle{alttree}{%
\renewenvironment{theglossary}%
{\def\@gls@prevlevel{-1}%
\mbox{}\par}%
{\par}%
\renewcommand*{\glossaryheader}{}%
\renewcommand*{\glsgroupheading}[1]{}%
\renewcommand{\glossentry}[2]{%
\ifnum\@gls@prevlevel=0\relax
\else
\glsmasurewidth{\glstreeindent}{\glstreenamefmt{\@glswidestname\space}}%
\fi
\hangindent\glstreeindent
\parindent\glstreeindent
\makebox[Opt][r]{\glstreenamebox{\glstreeindent}{%
\glstentryitem{##1}\glstreenamefmt{\glstarget{##1}{\glossentryname{##1}}}}}%
\ifglshassymbol{##1}{(\glossentrysymbol{##1})\space}{}%
\glossentrydesc{##1}\glspostdescription \space ##2\par
\def\@gls@prevlevel{0}%
}%
\renewcommand{\subglossentry}[3]{%
\ifnum##1=1\relax
\glssubentryitem{##2}%
\fi
\ifnum\@gls@prevlevel=##1\relax
\else
\@ifundefined{\@glswidestname\romannumeral##1}{%
\glsmasurewidth{\gls@tmplen}{\glstreenamefmt{\@glswidestname\space}}{%
\glsmasurewidth{\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}{%
        \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
            \@glswidestname\space}}}%
        \glsmeasurewidth{\glstreeindent}{\glstreenamefmt{%
            \csname @glswidestname\romannumeral\@gls@prevlevel
            \endcsname\space}}}%
        \addtolength\parindent{-\glstreeindent}%
        \setlength\glstreeindent\parindent
    \fi
\fi
\hangindent\glstreeindent
\makebox[0pt][r]{\glstreenamebox{\gls@tmplen}{%
    \glstreenamefmt{\glstarget{##2}{\glossentryname{##2}}}}}%
\ifglshassymbol{##2}{(\glossentrysymbol{##2})\space}{}%
\glossentrydesc{##2}\glspostdescription\space ##3\par
\def\@gls@prevlevel{##1}%
}%
\renewcommand*{\glsgroupskip}{\ifglsnogroupskip\else\indexspace\fi}%
}
\newglossarystyle{almtreegroup}{%
    \setglossarystyle{almtree}%
    \renewcommand{\glsgroupheading}[1]{\par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt{\glsgetgrouptitle{##1}}%
        \par\indexspace}%
}
\newglossarystyle{almtreehypergroup}{%
    \setglossarystyle{almtree}%
    \renewcommand*{\glossaryheader}{%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreenavigationfmt{\glsnavigation}\par\indexspace}%
    \renewcommand*{\glsgroupheading}[1]{%
        \par
        \def\@gls@prevlevel{-1}%
        \hangindent0pt\relax
        \parindent0pt\relax
        \glstreegroupheaderfmt
        {\glsnavhypertarget{##1}{\glsgetgrouptitle{##1}}}\par
        \indexspace}}

```

Change History

1.01 – 2007-05-17	
General: Added range facility in format key	134
\writeist: Added spaces after \delimN and \delimR in ist file	189
1.04 – 2007-08-03	
General: Added \glstextformat	122
1.05 – 2007-08-10	
\glossarysection: added	
\@mkboth to	
\glossarysection	58
\gls@defglossaryentry:	
Changed the default value of the sort key to just the value of the name key	104
1.07 – 2007-09-13	
\@gls@link: fixed bug caused by \theglentrycounter setting the page number too soon	132
\glsadd: fixed bug caused by \theglentrycounter setting the page number too soon	186
1.08 – 2007-10-13	
General: Added babel support	52
altlistgroup: changed altlistgroup style to use \glsgetgrouptitle	308
listgroup: changed listgroup style to use \glsgetgrouptitle	307
1.1 – 2008-02-22	
General: amsgen now loaded (\new@ifnextchar needed)	6
\@glossarysection: numbered sections and auto label added	60
\@gls@tmpb: changed \toksdef to \newtoks	136
\@gls@toc: numberline added	61
\@p@glossarysection: numbered sections and auto label added	60
numberedsection:	
numberedsection package option added	11
numberline: numberline option added	10
\setglossarysection: new	59
translate: translate option added	33
1.12 – 2008-03-08	
General: added check for \hypertarget separate to \hyperlink (memoir defines \hyperlink but not \hypertarget)	144
\@GLSpl: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	150
\@GLspl0: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	149
\@glspl0: now uses \glentrydescplural and \glentrysymbolplural instead of \glentrydesc and \glentrysymbol	148
descriptionplural: new	84
\gls@defglossaryentry:	
Changed default first plural to be first key with s appended (was text key with s appended)	103
descriptionplural support added	103
symbolplural support added	103
\Glsentrydescplural: New	178
\glentrydescplural: New	178
\Glsentrysymbolplural: New	179
\glentrysymbolplural: New	179
symbolplural: new	85
1.13 – 2008-05-10	
General: fixed bug that ignored 3rd parameter	151–160
\ACRfullpl: new	273
\Acrfullpl: new	273
\acrfullpl: new	273
\acrpluralsuffix: New	270
\gls@defglossaryentry:	
Changed default first value	103

Changed default firstplural value	103	\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	149
Removed restriction on only using \newglossaryentry in the preamble	108	\@gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	146
\newacronym: Removed restriction on only using \newacronym in the preamble	269	\@glsdisp: Test glossary type is \acronymtype in addition to checking if footnote option has been used	150
1.14 – 2008-06-17		\@GLspl@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	148
General: added nonumberlist key to \printglossary	254	\@gls@target: raised the hypertarget so the target text doesn't scroll off the top of the page	144
added numberedsection key to \printglossary	253	\@gls@defglossaryentry: Changed def to let	103
\@gls@hypergroup: new	302	1.17 – 2008-12-26	
\firstacronymfont: new	274	General: added xindy support	36
\glsautoprefix: new	11	\@do@esc@wrglossary: new	217
\glsnavhyperlink: changed \edef to \protected@edef	300	\@do@seeglossary: new	221
\glsnavhypertarget: added write to aux file	301	\@glo@storeentry: new	110
\glsnavigation: changed to only use labels for groups that are present	302	\@gls@glossary: changed definition to use \index instead of \@index	212
1.15 – 2008-08-15		\@glsdefaultplural: new	89
\@gls@link: added \glslabel	132	\@glsdefaultsort: new	89
\@gls@defglossaryentry: check for \@glo@first in description	107	\@gls@hypernumber: new	266
check for \@glo@text in symbol	108	\@glsnoname: new	88
\@gls@hypergroup: new	302	\@glsnonextpages: new	255
\@glsnavhypertarget: added check if rerun required	301	\@gls@defglossaryentry: added nonumberlist key	104
\@glssettoctitle: new	51	added parent key	104
\printglossary: changed the way the TOC title is set	224	added see key	104
1.16 – 2008-08-27		Stored main part of entry format when entry is defined	108
\@GLS@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	147	\@gls@suffixF: new	56
\@GLSpl: Test glossary type is \acronymtype in addition to checking if footnote option has been used	150	\@gls@suffixFF: new	56
\@Gls@: Test glossary type is \acronymtype in addition to checking if footnote option has been used	146	\@gls@wrglossary: modified to allow for xindy support	213
		\@gls@hyperlink: new	185
		\@gls@hypernumber: modified to allow material to be attached to location	266

\glsnavhyperlink: replaced	1.46 – 2021-09-20
\hyperlink to \glslink	General: changed \edef to
\glsnavhypertarget: replaced	\protected@edef 465
\hypertarget with	\@printglossary: changed \xdef
\@gls@target 301	to \protected@xdef 226
\glssee: new 222	\gls@defglossaryentry: changed
\glsseeformat: new 222	\xdef to
\glsSetSuffixF: new 56	\protected@xdef 105, 108
\glsSetSuffixFF: new 56	\glsaddprotectedpagefmt:
\ifglxindy: new 36	changed \eappto to
\istfilename: added xindy	\protected@eappto 215
support 55	2.01 – 2009 May 30
\newglossarystyle: made	\@gls@link: moved
\newglossarystyle long 265	\@do@wrglossary before term
nonumberlist: new 86	is displayed to prevent
\nopostdesc: new 54	unwanted whatsit 132
order: order package option	\forallglossaries: replaced
added 36	\ifthenelse with \ifx 70
parent: new 86	\forglsentries: replaced
\printglossary: added check to	\ifthenelse with \ifx 71
determine if \printglossary	\glsdefmain: new 21
is already defined 224	\glsdescwidth: changed
added print language to aux	\linewidth to \hsize 310, 335
file 224	\glslistdottedwidth: changed
see: new 85	\linewidth to \hsize 309
\writeist: added xindy support 189	\glspagelistwidth: changed
1.18 – 2009-01-14	\linewidth to \hsize 310, 335
\@gls@loadlist: new 14	nomain: added nomain package
\@gls@loadlong: new 13	option 21
\@gls@loadsuper: new 14	\writeist: removed item_02 -
\@gls@loadtree: new 14	no such makeindex key 193
\gls@defglossaryentry:	2.02 – 2007-07-13
Changed default value of sort	\@printglossary: suppressed
to \glsdefaultsort 104	warning globally rather than
moved sort sanitization to	locally 227
\newglossaryentry 108	2.02 – 2009-07-13
\glstarget: new 258	\glossarysection: changed
nolist: new 14	\@mkboth to \glossarymark 58
nolong: new 13	\gls@defglossarymark: New 59
nostyles: new 14	2.03 – 2009-09-23
nosuper: new 14	\@GLS@: Added check for
notree: new 14	hyperfirst 147
\oldacronym: new 269	\@GLSpl: Added check for
sort: moved sanitization to	hyperfirst 150
\newglossaryentry 84	\@GLS@: Added check for
1.19 – 2009-03-02	hyperfirst 146
\gls@clearpage: new 61	\@GLSpl@: Added check for
\glsdisp: new 150	hyperfirst 149
	\@GLS@: Added check for
	hyperfirst 146

\@gls@link: new	130	\gls@defglossaryentry: added	
\@gls@link: added \leavevmode	132	user1-6 keys	104
Moved entry existence check to		\glsadd: fixed bug that ignored	
avoid duplicate code	132	counter	186
\@glsdisp: Added check for		\Glsentryuseri: new	181
hyperfirst	150	\glsentryuseri: new	181
\@glspl@: Added check for		\Glsentryuserii: new	181
hyperfirst	148	\glsentryuserii: new	181
\gls glossarymark: Added check		\Glsentryuseriii: new	181
to see if it's already defined .	59	\glsentryuseriii: new	181
hyperfirst: new	34	\Glsentryuseriv: new	182
2.04 – 2009-11-10		\glsentryuseriv: new	181
\@GLS@: Changed test to check if		\Glsentryuserv: new	182
glossary type has been		\glsentryuserv: new	182
identified as a list of		\Glsentryuservi: new	182
acronyms	147	\glsentryuservi: new	182
\@GLSpl: Changed test to check if		\ns@newglossary: added check to	
glossary type has been		determine if	
identified as a list of		\gls@(<type>@display and	
acronyms	150	\gls@(<type>@displayfirst	
\@Gls@: Changed test to check if		have been defined.	81
glossary type has been		\SetAcronymLists: new	24
identified as a list of		\SetDefaultAcronymDisplayStyle:	
acronyms	146	new	289
\@Glspl@: Changed test to check		\SetDefaultAcronymStyle: new	289
if glossary type has been		\SetDescriptionAcronymDisplayStyle:	
identified as a list of		new	290
acronyms	149	\SetDescriptionDUAAcronymDisplayStyle:	
\@glossaryentryfield: new ..	109	new	290
\@glossarysubentryfield: new	109	\SetDescriptionFootnoteAcronymDisplayStyle:	
\@gls@: Changed test to check if		new	290
glossary type has been		\SetDUADisplayStyle: new ...	291
identified as a list of		\SetFootnoteAcronymDisplayStyle:	
acronyms	146	new	290
\@glsacronymlists: new	22	\SetSmallAcronymDisplayStyle:	
\@glsdisp: Changed test to check		new	291
if glossary type has been		2.05 – 2010-02-06	
identified as a list of		\@glsdisp: Added closing brace.	
acronyms	150	Patch provided by Sergiu	
\@glspl@: Changed test to check		Dotenco	150
if glossary type has been		Removed spurious brace. Patch	
identified as a list of		provided by Sergiu Dotenco	150
acronyms	148	\writeist: Added \string	
\@newglossaryentryposthook:		before opening and closing	
new	109	braces. Patch provided by	
\@newglossaryentryprehook:		Sergiu Dotenco	194
new	109	2.06 – 2010-06-14	
acronymlists: new	24	\altnewglossary: new	81
\DeclareAcronymList: new ...	23	\CustomAcronymFields: new ..	292
\DefineAcronymSynonyms: new .	287	\CustomNewAcronymDef: new ..	292

\SetCustomDisplayStyle: new .	292	added \glsnextpages	226
\SetCustomStyle: new	292	make toctitle default to title .	226
2.07 – 2010-07-10		\@xdyattributelist: new	62
General: glsadd format key stored		\acrfootnote: new	289
in \@glsnumberformat (was		\ACRfull: added starred version	273
mistakenly stored in		\Acrfull: added starred version	272
\@glo@format)	185	\acrfull: added starred version	272
3.0 – 2010-07-12		\ACRfullpl: added starred	
\@makeglossary: Added check for		version	273
savewrites	199	\Acrfullpl: added starred	
\gls@wrglossary: modified to		version	273
take into account savewrites	213	\acrfullpl: added starred	
3.0 – 2010/03/31		version	273
\@set@glo@numformat: added 4th		\acrlinkfootnote: new	289
argument	134	\acrnolinkfootnote: new	290
3.0 – 2011-04-02		compatible-2.07:	
General: added prefix to		compatible-2.07 option added	42
hyperlink	267	counter: replaced \@ifundefined	
etoolbox now loaded	6	with \ifcsundef	85
replaced \@ifundefined with		counterwithin: new	16
\ifcsundef	50, 52, 128, 253	entrycounter: new	16
\@do@esc@wrglossary: added		glossaryentry: new	15
check for hyper location		\glossarysection: replaced	
prefix	219	\@ifundefined with	
modified to use new format .	217	\ifcsundef	58
\@glossarysec: replaced		\glossarystyle: replaced	
\@ifundefined with		\@ifundefined with	
\ifcsundef	10	\ifcsundef	265
\@do@seeglossary: Sanitize and		glossarysubentry: new	16
escape cross-referencing		\gls@codepage: replaced	
information	221	\@ifundefined with	
\@gls@counterwithin: new . . .	16	\ifcsundef	37
\@gls@ifinlist: new	62	\gls@defglossaryentry: added	
\@gls@link: added		\@gls@defsort	108
\@gls@saveentrycounter .	132	added short and long keys . .	104
added \@gls@setsort	132	replaced \@ifundefined with	
\@gls@saveentrycounter: new .	133	\ifcsundef	104
\@gls@setupsort@def: new . . .	18	\gls@doclearpage: replaced	
\@gls@setupsort@standard: new	17	\@ifundefined with	
\@gls@setupsort@use: new . . .	19	\ifcsundef	60
\@gls@xdy@locationlist: new .	65	\glsadd: added	
\@glslink: replaced		\@gls@saveentrycounter .	186
\@ifundefined with		\GlsAddXdyCounters: new	62
\ifcsundef	144	\glsentrycounterlabel: new .	257
\@glsnextpages: new	255	\glsentryitem: new	257
\@print@glossary: replaced		\Glsentrylong: new	183
\@ifundefined with		\glsentrylong: new	183
\ifcsundef	227	\Glsentrylongpl: new	183
\@printglossary: added		\glsentrylongpl: new	183
\currentglossary	226	\Glsentryshort: new	182

<code>\glstryshort: new</code>	182	replaced <code>\@ifundefined</code> with	
<code>\Glsentryshortpl: new</code>	183	<code>\ifcsundef</code>	81
<code>\glstryshortpl: new</code>	182	<code>\oldacronym: replaced</code>	
<code>\glsgrouptitle: replaced</code>		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	269
<code>\ifcsundef</code>	262	<code>\printglossary: replaced</code>	
<code>\glsglossarymark: replaced</code>		<code>\@ifundefined</code> with	
<code>\@ifundefined</code> with		<code>\ifcsundef</code>	224
<code>\ifcsundef</code>	59	<code>savewrites: new</code>	42
<code>\glshyperlink: changed default</code>		<code>see: added</code>	
from <code>\glstryname</code> to		<code>\@glo@seeautonumberlist</code> .	85
<code>\glstrytext</code>	185	<code>seeautonumberlist: new</code>	13
<code>\glshypernumber: replaced</code>		<code>\SetDescriptionFootnoteAcronymDisplayStyle:</code>	
<code>\@ifundefined</code> with		expanded options link	
<code>\ifcsundef</code>	266	options	290
<code>\glstnumberformat: replaced</code>		<code>\setentrycounter: added</code>	
<code>\@ifundefined</code> with		optional argument	264
<code>\ifcsundef</code>	57	<code>short: new</code>	88
<code>\glssrefentry: new</code>	256	<code>shortplural: new</code>	88
<code>\glssresetsubentrycounter:</code>		<code>\showacronymlists: new</code>	298
new	256	<code>\showglocounter: new</code>	294
<code>\glssseeitem: hyperlink uses</code>		<code>\showglodesc: new</code>	295
<code>\glssseeitemformat</code> instead of		<code>\showglodescplural: new</code>	296
<code>\glstryname</code>	223	<code>\showglofirst: new</code>	293
<code>\glssseeitemformat: new</code>	223	<code>\showglofirstpl: new</code>	294
<code>\glssortnumberfmt: new</code>	18	<code>\showgloflag: new</code>	297
<code>\glssstepentry: new</code>	256	<code>\showgloindex: new</code>	297
<code>\glssstepsubentry: new</code>	256	<code>\showglolevel: new</code>	293
<code>\glsssubentrycounterlabel:</code>		<code>\showgloname: new</code>	295
new	257	<code>\showgloparent: new</code>	293
<code>\glsssubentryitem: new</code>	257	<code>\showgloplural: new</code>	293
<code>\ifglossaryexists: replaced</code>		<code>\showglosort: new</code>	296
<code>\@ifundefined</code> with		<code>\showglossaries: new</code>	298
<code>\ifcsundef</code>	71	<code>\showglossarycounter: new</code>	298
<code>\ifglstryexists: replaced</code>		<code>\showglossaryentries: new</code>	299
<code>\@ifundefined</code> with		<code>\showglossaryin: new</code>	298
<code>\ifcsundef</code>	73	<code>\showglossaryout: new</code>	298
<code>\istfile: deprecated</code>	211	<code>\showglossarytitle: new</code>	298
<code>long: new</code>	88	<code>\showglosymbol: new</code>	296
<code>longplural: new</code>	88	<code>\showglosymbolplural: new</code>	296
<code>\newglossaryentry: replaced</code>		<code>\showglotext: new</code>	293
<code>\DeclareRobustCommand</code> with		<code>\showglotype: new</code>	294
<code>\newrobustcmd</code>	91	<code>\showglouser: new</code>	294
<code>\newglossarystyle: replaced</code>		<code>\showglouserii: new</code>	294
<code>\@ifundefined</code> with		<code>\showglouseriii: new</code>	295
<code>\ifcsundef</code>	265	<code>\showglouseriv: new</code>	295
<code>nonumberlist: now boolean</code>	86	<code>\showglouserv: new</code>	295
<code>\ns@newglossary: added</code>		<code>\showglouservi: new</code>	295
<code>\@gls@defsortcount</code>	81	<code>sort: new</code>	17
		<code>subentrycounter: new</code>	16

theglossary: replaced	\glsname: made robust	155
\@ifundefined with	\GLSpl: made robust	149
\ifcsundef	\Glspl: made robust	148
\writeist: added xindy-only	\glspl: made robust	147
macro definitions to glossary	\GLSplural: made robust	154
open tag	\GLSsymbol: made robust	159
modified to support new	\Glsymbol: made robust	158
format	\glsymbol: made robust	158
189	\GLSsymbolplural: made robust	160
3.01 – 2011-04-12	\Glsymbolplural: made robust	159
General: made robust	\glsymbolplural: made robust	159
\@glswritefiles: added check	\GLstext: made robust	152
for empty glossaries	\glstext: made robust	151
211	\GLSuseri: made robust	161
\ACRfull: made robust	\Glsuseri: made robust	160
272	\glsuseri: made robust	160
\Acrfull: made robust	\GLSuserii: made robust	162
272	\Glsuserii: made robust	161
\acrfullformat: removed	\glsuserii: made robust	161
\acronymfont as it should	\GLSuseriii: made robust	163
already be set in the second	\Glsuseriii: made robust	162
argument.	\glsuseriii: made robust	162
271	\GLSuseriv: made robust	164
\ACRfullpl: made robust	\Glsuseriv: made robust	163
273	\glsuseriv: made robust	163
\Acrfullpl: made robust	\GLSuserv: made robust	165
273	\Glsuserv: made robust	164
\acrfullpl: made robust	\glsuserv: made robust	164
273	\GLSuservi: made robust	166
\ACRlong: made robust	\Glsuservi: made robust	165
171	\glsuservi: made robust	165
\Acrlong: made robust		
171		
\acrlong: made robust		
170		
\ACRlongpl: made robust		
173		
\Acrlongpl: made robust		
173		
\acrlongpl: made robust		
172		
\ACRshort: made robust		
167		
\Acrshort: made robust		
167		
\acrshort: made robust		
166		
\ACRshortpl: made robust		
169		
\Acrshortpl: made robust		
169		
\acrshortpl: made robust		
168		
\GLs: made robust		
146		
\glsadd: made robust		
186		
\glsaddall: made robust		
186		
\GLSdesc: made robust		
157		
\GLSdesc: made robust		
156		
\glsdesc: made robust		
156		
\GLSdescplural: made robust		
158		
\Glsdescplural: made robust		
157		
\glsdescplural: made robust		
157		
\glsfirst: made robust		
152		
\GLSfirstplural: made robust		
155		
\Glsfirstplural: made robust		
154		
\glsfirstplural: made robust		
154		
\glslink: made robust		
130		
\GLSname: made robust		
156		
\Glsname: made robust		
155		
	\glsnumlistlastsep: new	185
	\glsnumlistsep: new	185
	3.02 – 2012-05-21	
	\@do@wrglossary: changed	
	\@glslocref to	
	\theglsentrycounter	220
	\@do@wrglossary: changed	
	\@do@wrglossary to test for	
	indexonlyfirst option; put old	
	\@do@wrglossary code into	
	\@do@wrglossary	214
	\@gls@missingnumberlist: new	89
	\@glswritefiles: added check	
	for existence of token in case	
	\makeglossaries has been	
	omitted	211
	\@printglossary: add a way to	
	fetch current entry label	226

<code>\gls@defglossaryentry</code> : added numberlist element	107	<code>\gls@postinline</code> : replaced “.” with <code>\gls@postdescription</code>	305
<code>\gls@save@numberlist</code> : new	223	<code>index</code> : added check for <code>glsnogroupskip</code>	412
<code>\gls@wrglossary</code> : added check for glossary file defined	213	<code>list</code> : added check for <code>glsnogroupskip</code>	307
<code>\glsdisplaynumberlist</code> : new	184	<code>long</code> : added check for <code>glsnogroupskip</code>	311
<code>\glsentrycounter</code> : set default value	133	<code>long3col</code> : added check for <code>glsnogroupskip</code>	312
<code>\Glsentryfull</code> : fixed bug (replaced <code>\glsentryshortpl</code> with <code>\glsentryshort</code>)	183	<code>long4col</code> : added check for <code>glsnogroupskip</code>	314
<code>\glsentryfullpl</code> : fixed bug (replaced <code>\glsentryshort</code> with <code>\glsentryshortpl</code>)	183	<code>longragged</code> : added check for <code>glsnogroupskip</code>	322
<code>\glsentrynumberlist</code> : new	184	<code>longragged3col</code> : added check for <code>glsnogroupskip</code>	324
<code>\glsmoveentry</code> : new	109	<code>nogroupskip</code> : new	15
<code>\glsresetentrycounter</code> : new	256	<code>nopostdot</code> : new	15
<code>\ifglshaschildren</code> : new	74	<code>super</code> : added check for <code>glsnogroupskip</code>	335
<code>\ifglshasparent</code> : new	75	<code>super3col</code> : added check for <code>glsnogroupskip</code>	337
<code>indexonlyfirst</code> : new	34	<code>super4col</code> : added check for <code>glsnogroupskip</code>	339
<code>\makeglossaries</code> : added list parser	204	<code>superragged</code> : added check for <code>glsnogroupskip</code>	343
<code>\renewglossarystyle</code> : new	265	<code>superragged3col</code> : added check for <code>glsnogroupskip</code>	345
<code>savenumberlist</code> : new	13	<code>tree</code> : added check for <code>glsnogroupskip</code>	413
<code>\showglossaryentries</code> : fixed misspelt command	299	<code>treenoname</code> : added check for <code>glsnogroupskip</code>	415
<code>\SmallNewAcronymDef</code> : fixed broken short and long plural	291		
<code>ucmark</code> : new	15		
3.03 – 2012/09/21		3.04 – 2012-11-11	
General: allow title to set <code>toctitle</code>	253	<code>altlist</code> : replaced <code>\newline</code> with paragraph break	308
<code>\@gls@sanitizesort</code> : new	27	3.04 – 2012-11-18	
<code>\@gls@setupsort@standard</code> : used <code>\@gls@sanitizesort</code>	17	General: Added check for doc package	6
<code>\@printglossary</code> : allow title to override default <code>toctitle</code>	225	added <code>datatool</code> -base as a required package	54
<code>altlongragged4col</code> : added check for <code>glsnogroupskip</code>	325	added local key	129
<code>altsuperragged4col</code> : added check for <code>glsnogroupskip</code>	346	<code>\@do@wrglossary</code> : changed <code>\theglsentrycounter</code> back to <code>\@glslocref</code>	220
<code>alttree</code> : added check for <code>glsnogroupskip</code>	418	<code>\@do@esc@wrglossary</code> : modified to compensate for possible incorrect page number	218
<code>\glsinlinedescformat</code> : new	305	<code>\@gls@esc@csdq</code> : unsanitize <code>\gls@numberpage</code> , <code>\gls@alphapage</code> ,	
<code>\glsinlineemptydescformat</code> : new	305		
<code>\glsinlinenameformat</code> : new	305		
<code>\glsinlinepostchild</code> : new	305		
<code>\glsinlinesubdescformat</code> : new	305		
<code>\glsinlinesubnameformat</code> : new	305		

\gls@Alphpage and \gls@romanpage	135	\findrootlanguage: Obsoleted	69
\@print@glossary: Moved aux write to end of document to prevent unwanted whatsit occurring here.	227	3.07 – 2013-07-05 \@gls@link: fixed bug that failed to find entry in list	132
\gls@Alphpage: new	214	\glossarypreamble: modified to work with \setglossarypreamble	57
\gls@alphpage: new	214	\gls@docclearpage: added check for openright	60
\gls@disablepagerefexpansion: new	214	\glspostdescription: Added spacefactor code	14
\gls@numberpage: new	215	\GlsSetXdyCodePage: Added check for fontspec	70
\gls@protected@pagefmts: new	214	\SetDescriptionAcronymDisplayStyle: now using \glsdoparenifnotempty	290
\gls@romanpage: new	215	\setglossarypreamble: new	58
\glsdefmain: added check for doc package	21	3.08a – 2013-08-30 altlist: updated altlist style to use \glossentry and \subglossentry	308
\glsorg@endtheglossary: new	6	inline: updated inline style to use \glossentry and \subglossentry	304
\glsorg@theglossary: new	6	list: updated list style to use \glossentry and \subglossentry	307
\PrintChanges: new	6	listdotted: updated listdotted style to use \glossentry and \subglossentry	309
3.05 – 2013-04-21 \@do@esc@wrglossary: add Roman case. Fixed bugs in the else statements	218	3.08a – 2013-09-28 General: added nogroupskip key to \printglossary	254
\@gls@link: added check for “nohypertypes”	132	removed definition of \@glossaryentryfield	454
\gls@protected@pagefmts: added Roman to list	214	removed definition of \@glossarysubentryfield	454
\gls@Romanpage: new	215	\@glo@storeentry: no longer need to check for special characters in any of the fields other than sort	110
\glsgetgrouplabel: fixed bug (typo in \equal)	263	updated for \glossentry	110
\glsmcols: new	329	\@glossaryentryfield: switched to \glossentry	109
mcolalttree: replaced ‘2’ with \glsmcols	333	\@glossarysubentryfield: switched to \subglossentry	109
mcolindex: replaced ‘2’ with \glsmcols	329	\@gls@sanitizedesc: deprecated	26
mcolindexspannav: replaced ‘2’ with \glsmcols	330	altlongragged4col: updated to use \glossentry and \subglossentry	325
mcoltree: replaced ‘2’ with \glsmcols	330		
mcoltreenoname: replaced ‘2’ with \glsmcols	332		
mcoltreespannav: replaced ‘2’ with \glsmcols	331		
\nopostdesc: made robust	54		
3.05 – 2013/04/21 \@gls@nohyperlist: new	24		
\GlsDeclareNoHyperList: new	24		
nohypertypes: new	24		
3.06 – 2013/06/17 \@xdy@main@language: Changed back to using \language	36		

alttree: updated to use		superragged: updated to use	
\glossentry and		\glossentry and	
\subglossentry	417	\subglossentry	342
\compatibleglossentry: new ..	258	tree: updated to use	
\compatiblesubglossentry:		\glossentry and	
new	260	\subglossentry	413
\glossaryentryfield:		3.09a – 2013-10-09	
deprecated	260	\@gls@assign@symbolplural@field:	
\glossarystyle: deprecated ..	265	new	27
\glossentry: new	258	\@gls@default@value: new ..	84
\Glossentrydesc: new	259	\Glsentrydesc: made robust ..	177
\glossentrydesc: new	259	\Glsentrydescplural: made	
\Glossentryname: new	258	robust	178
\glossentryname: new	258	\Glsentryfirst: made robust ..	179
\Glossentrysymbol: new	260	\Glsentryfirstplural: made	
\glossentrysymbol: new	259	robust	180
\gls@assign@desc@field: new ..	26	\Glsentryfull: made robust ..	183
\gls@assign@descplural@field:		\Glsentryfullpl: made robust ..	184
new	26	\Glsentrylong: made robust ..	183
\gls@assign@field: new	91	\Glsentrylongpl: made robust ..	183
\gls@ifnotmeasuring: new ..	112	\Glsentryname: made robust ..	176
\glsaddallunused: new	187	\Glsentryplural: made robust ..	178
\glsexpandfields: new	91	\Glsentryshort: made robust ..	182
\glsnoexpandfields: new	91	\Glsentryshortpl: made robust ..	183
\glssee: made robust	222	\Glsentrysymbol: made robust ..	179
\glsseeformat: made robust ..	222	\Glsentrysymbolplural: made	
\glsseeitem: made robust	223	robust	179
\glsseelist: made robust	222	\Glsentrytext: made robust ..	178
\ifglshdescsuppressed: new ..	75	\Glsentryuseri: made robust ..	181
\ifglshasdesc: new	75	\Glsentryuserii: made robust ..	181
\ifglshassymbol: new	76	\Glsentryuseriii: made robust ..	181
index: added paragraph break at		\Glsentryuseriv: made robust ..	182
end of environment	411	\Glsentryuserv: made robust ..	182
updated to use \glossentry		\Glsentryuservi: made robust ..	182
and \subglossentry	411	\glstextup: new	270
long: updated to use		\ifglshassymbol: changed test	
\glossentry and		to check for	
\subglossentry	311	\@gls@default@symbol	76
longragged: updated to use		3.10a – 2013-09-28	
\glossentry and		\gls@assign@type@field: new ..	26
\subglossentry	322	3.10a – 2013-10-13	
longragged3col: updated to use		\@gls@keymap: new	94
\glossentry and		\@gls@provide@newglossary:	
\subglossentry	323	new	79
\setglossarystyle: new	264	\@gls@writedef: new	94
\setglossentrycompatibility:		\@glsdefaultplural: Obsolete ..	89
new	260	\@glsnodesc: new	88
\subglossentry: new	261	\@print@glossary: Added	
		providecommand code to aux	
		file	228

<code>\gls@defglossaryentry:</code>	
Changed to using	
<code>\@gls@default@value</code> .	103, 104
new	103
<code>\glswritedefhook:</code> new	102
<code>\makeglossaries:</code> Added	
providecommand code to aux	
file	203
<code>\new@glossaryentry:</code> new	92
<code>\ns@newglossary:</code> added	
<code>\@gls@provide@newglossary</code>	81
3.11a – 2013-10-15	
General: added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	166–174
changed to just use	
<code>\Glsentrydescplural</code>	158
changed to just use	
<code>\glsentrydescplural</code>	157, 158
changed to just use	
<code>\Glsentrydesc</code>	157
changed to just use	
<code>\glsentrydesc</code>	156, 157
changed to just use	
<code>\Glsentryfirstplural</code>	155
changed to just use	
<code>\glsentryfirstplural</code>	154, 155
changed to just use	
<code>\Glsentryfirst</code>	153
changed to just use	
<code>\glsentryfirst</code>	152, 153
changed to just use	
<code>\Glsentryname</code>	156
changed to just use	
<code>\glsentryname</code>	155, 156
changed to just use	
<code>\Glsentryplural</code>	154
changed to just use	
<code>\glsentryplural</code>	153, 154
changed to just use	
<code>\Glsentrysymbolplural</code>	160
changed to just use	
<code>\glsentrysymbolplural</code>	159, 160
changed to just use	
<code>\Glsentrysymbol</code>	159
changed to just use	
<code>\glsentrysymbol</code>	158, 159
Changed to just use	
<code>\Glsentrytext</code>	152
changed to just use	
<code>\glsentrytext</code>	151
changed to just use	
<code>\Glsentryuseriii</code>	163
changed to just use	
<code>\glsentryuseriii</code>	162, 163
changed to just use	
<code>\Glsentryuserii</code>	162
changed to just use	
<code>\glsentryuserii</code>	161, 162
changed to just use	
<code>\Glsentryuseriv</code>	164
changed to just use	
<code>\glsentryuseriv</code>	163, 164
changed to just use	
<code>\Glsentryuseri</code>	161
changed to just use	
<code>\glsentryuseri</code>	160, 161
changed to just use	
<code>\Glsentryuservi</code>	166
changed to just use	
<code>\glsentryuservi</code>	165, 166
changed to just use	
<code>\Glsentryuserv</code>	165
changed to just use	
<code>\glsentryuserv</code>	164, 165
Now requires <code>textcase</code>	5
<code>\@ACRlong:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	449
<code>\@ACRshort:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	448
<code>\@Acrlong:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	449
<code>\@Acrshort:</code> added <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glsinsert</code> and	
<code>\glscustomtext</code>	447
<code>\@GLS@:</code> add <code>\glslabel</code> ,	
<code>\glsifplural</code> , <code>\glscapscase</code> ,	
<code>\glscustomtext</code> and	
<code>\glsinsert</code>	147
change to using <code>\glsentryfmt</code>	
style commands	147

removed \MakeUppercase (now moved to \glentryfmt) ..	147	\@glsp1@: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	148
\@GLSpl: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	149	change to using \glentryfmt style commands	150
change to using \glentryfmt style commands	150	removed \MakeUppercase as now dealt with in \glentryfmt	150
\@Gls@: add \glslifplural, \glscapscase, \glscustomtext and \glinsert	146	change to using \glentryfmt style commands	146
removed \makefirsttuc (now dealt with in \glentryfmt)	146	\@GLspl@: add \glslifplural, \glscapscase, \glscustomtext and \glinsert	149
\@GLspl@: add \glslifplural, \glscapscase, \glscustomtext and \glinsert	149	change to using \glentryfmt style commands	149
removed \makefirsttuc (now dealt with in \glentryfmt)	149	\@acrlong: added \glslabel, \glslifplural, \glscapscase, \glinsert and \glscustomtext	448
\@acrlong: added \glslabel, \glslifplural, \glscapscase, \glinsert and \glscustomtext	447	\@acrlong: added \glslabel, \glslifplural, \glscapscase, \glinsert and \glscustomtext	447
\@Gls@: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	145	change to using \glentryfmt style commands	146
\@Gls@noexpand@fields: Fixed bug expand replaced with noexpand	90	\@Glsdisp: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	150
\@Glsdisp: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	150	change to using \glentryfmt style commands	150
\@glsp1@: add \glslabel, \glslifplural, \glscapscase, \glscustomtext and \glinsert	148	change to using \glentryfmt style commands	148
acronymlists: replaced \@addtoacronymlists with \DeclareAcronymList	24	compatible-3.07: new	42
\defglssdisplay: obsoleted	127	\defglssdisplayfirst: obsoleted	127
\defglssentryfmt: new	79	\forglssentries: replaced \ifx with \ifdefempty	71
\glssassign@desc: new	102	\glss@defglssaryentry: Fixed default counter if none supplied	107
\glssdoentryfmt: new	79	\glssdisplay: obsoleted	127
\glssdisplayfirst: obsoleted ..	127	\glssdisplayfirst: obsoleted ..	127
\glssgenentryfmt: new	122	\glssgetgrouptitle: Added check in case non-Latin alphabet in use	262
\glssglossarymark: replaced \MakeUppercase with \mfirsttucMakeUppercase ..	59	\glssnavigation: switched to using \@glss@getgrouptitle	303
\glssnavigation: switched to using \@glss@getgrouptitle	303	\ifglsshasdesc: replaced \ifdefempty with \ifcsemtty	75
\ifglsshasdesc: replaced \ifdefempty with \ifcsemtty	75	\ifglsshaslong: new	76
\ifglsshaslong: new	76	\ifglsshasshort: new	76
\ifglsshasshort: new	76	\ifglsshasymbol: replaced \ifdefempty with \ifcsemtty	76
\ifglsshasymbol: replaced \ifdefempty with \ifcsemtty	76	\ifglssused: replaced \ifthenelse with \ifbool ..	73
\ifglssused: replaced \ifthenelse with \ifbool ..	73	\longnewglossaryentry: new ..	102
\longnewglossaryentry: new ..	102	\ns@newglossary: replaced \glssdisplay and \glssdisplayfirst with \glssentryfmt	81
\ns@newglossary: replaced \glssdisplay and \glssdisplayfirst with \glssentryfmt	81	numbers: new	44
numbers: new	44	\SetCustomDisplayStyle: updated to use \defglssentryfmt	292
\SetCustomDisplayStyle: updated to use \defglssentryfmt	292		

\SetDefaultAcronymDisplayStyle:	\gls@assign@descplural@field:
changed to use	changed to use
\defglsentryfmt 289	\glssetnoexpandfield 26
\SetDescriptionAcronymDisplayStyle:	\gls@assign@name@field:
updated to use	changed to use
\defglsentryfmt 290	\glssetnoexpandfield 26
\SetDescriptionDUAAcronymDisplayStyle:	\gls@assign@type@field:
updated to use	changed to use
\defglsentryfmt 290	\glssetexpandfield 26
\SetDescriptionFootnoteAcronymDisplayStyle:	\gls@checkseeallowed: new .. 86
updated to use	\glsaddallunused: set default to
\defglsentryfmt 290	\@glo@types 187
\SetDUADisplayStyle: updated	\Glsentryfull: changed to use
to use \defglsentryfmt .. 291	\acrfullformat 183
\SetFootnoteAcronymDisplayStyle:	\glsentryfull: changed to use
updated to use	\acrfullformat 183
\defglsentryfmt 290	\Glsentryfullpl: changed to use
\SetSmallAcronymDisplayStyle:	\acrfullformat 184
updated to use	\glsentryfullpl: changed to use
\defglsentryfmt 291	\acrfullformat 183
\setupglossaries: new 49	\gls glossarymark: renamed
\showglo long: new 297	\ glossarymark to
\showglo short: new 296	\gls glossarymark to avoid
symbols: new 44	conflict with memoir 59
3.12a – 2013-10-16	\glsprestandardsort: new ... 17
\gls@defglossaryentry: added	\glssetexpandfield: new 26
\glslabel 103	\glssetnoexpandfield: new .. 26
\glsaddkey: new 96	long: switched to
3.13a – 2013-11-05	\tabularnewline 311
General: changed \ glossarymark	long3col: switched to
to \gls glossarymark 59	\tabularnewline 312
\@gls@assign@symbol@field:	long3colheader: switched to
changed to use	\tabularnewline 313
\glssetnoexpandfield 27	long3colheaderborder: switched
\@gls@assign@symbolplural@field:	to \tabularnewline 313
changed to use	long4col: switched to
\glssetnoexpandfield 27	\tabularnewline 314
\@gls@link: removed \relax .. 132	long4colheader: switched to
\@gls@notranslatorhook: new . 32	\tabularnewline 314
\@gls@setupsort@standard:	longheader: switched to
moved \@gls@santizesort to	\tabularnewline 311
\glsprestandardsort 17	longheaderborder: switched to
altsuper4colheader: switched to	\tabularnewline 312
\tabularnewline 340	see: added
altsuper4colheaderborder:	\gls@checkseeallowed 85
switched to	super: switched to
\tabularnewline 341	\tabularnewline 335
\gls@assign@desc@field:	super3col: switched to
changed to use	\tabularnewline 337
\glssetnoexpandfield 26	

super3colheader: switched to \tabularnewline	338	\@gls@indexdef: new	45
super4col: switched to \tabularnewline	339	\@gls@numbersdef: new	45
super4colheader: switched to \tabularnewline	339	\@gls@symbolsdef: new	44
super4colheaderborder: switched to \tabularnewline	340	\ACRfullfmt: new	271
superheader: switched to \tabularnewline	336	\Acrfullfmt: new	271
superheaderborder: switched to \tabularnewline	336	\ACRfullplfmt: new	272
ucmark: added check for memoir	15	\Acrfullplfmt: new	272
3.14a – 2013-11-12		\acronymentry: new	276
General: new	421	dua: new	282
\@glswritefiles: renamed \glswritefiles to \@glswritefiles and used “savewrites” option to set \glswritefiles	211	dua-desc: new	284
acronyms: new	22	footnote: new	284
compatible-2.07: added check for 2.07 before setting 3.07 compatibility	42	footnote-desc: new	286
\gls@defglossaryentry: added check for existence of default glossary	104	footnote-sc: new	286
set the default for firstplural to be the value of plural	106	footnote-sc-desc: new	287
\longprovideglossaryentry: new	103	footnote-sm: new	286
notranslate: new	33	footnote-sm-desc: new	287
\provideglossaryentry: new	91	\Genacrfullformat: new	127
xindygloss: new	37	\genacrfullformat: new	126
4.0 – 2013-11-14		\GenericAcronymFields: new	276
\gls@defglossaryentry: added check for first key	106	\Genplacrfullformat: new	127
super: fixed typo in \subglossentry (\glossentrydesc)	335	\genplacrfullformat: new	127
4.01 – 2013-11-16		\Glsentryfull: bug fix: added missing \acronymfont	183
General: fixed non-value options so that they can be passed to document class	12	\Glsentryfull: bug fix: added missing \acronymfont	183
4.02 – 2013-12-05		\Glsentryfullpl: bug fix: added missing \acronymfont	184
General: Removed \acronymfont	170–174	\Glsentryfullpl: bug fix: added missing \acronymfont	183
\@acrfull: now using \acrfullfmt	272	\glsgenacfmt: new	124
		\GlsUseAcrEntryDisplayStyle: new	277
		\GlsUseAcrStyleDefs: new	277
		index: new	45
		long-sc-short: new	279
		long-sc-short-desc: new	281
		long-short: new	277
		long-short-desc: new	280
		long-sm-short: new	279
		long-sm-short-desc: new	281
		\newacronymstyle: new	276
		numberedsection: added nameref option	11
		sanitize: fixed bug that caused an error here	32
		sc-short-long: new	280
		sc-short-long-desc: new	282
		\setacronymstyle: new	276

\SetGenericNewAcronym: new	274	moved \@gls@link@opts and \@gls@link@label to \@gls@link	132
short-long: new	279	\gls@writedef: added	
short-long-desc: new	281	\glsdetoklabel	94
sm-short-long: new	280	\@glsdisp: removed \glslabel (defined in \@gls@link)	150
sm-short-long-desc: new	282	\@glspl@: removed \glslabel (defined in \@gls@link)	148
xindynoglsnumbers: new	37	\@printglossary: added	
4.02 – 2013-13-05		\glsdetoklabel	226
\makeglossaries: made preamble only	205	\compatibleglossentry: added	
4.03 – 2014-01-17		\glsdetoklabel	429
General: changed default to \@empty instead of \@relax	42	\compatiblesubglossentry: added \glsdetoklabel	429
4.03 – 2014-01-20		dua: fixed bug in \@acrfullfmt	283
General: removed \glslabel (defined in \@gls@link)	166	fixed bug in \@acrfullplfmt	284
\@do@esc@wrglossary: added		fixed bug in \@acrfullplfmt	284
\glsdetoklabel	219	redefined to use accessibility information	459
\@do@noesc@wrglossary: added		dua-desc: commented spurious EOL	284
\glsdetoklabel	216	redefined to use accessibility information	461
\@ACRlong: removed \glslabel (defined in \@gls@link)	449	footnote: redefined to use accessibility information	462
\@ACRshort: removed \glslabel (defined in \@gls@link)	448	footnote-desc: redefined to use accessibility information	464
\@Acrlong: removed \glslabel (defined in \@gls@link)	449	footnote-sc: redefined to use accessibility information	463
\@Acrshort: removed \glslabel (defined in \@gls@link)	447	footnote-sc-desc: redefined to use accessibility information	464
\@GLS@: removed \glslabel (defined in \@gls@link)	147	footnote-sm: redefined to use accessibility information	464
\@GLSpl: removed \glslabel (defined in \@gls@link)	149	footnote-sm-desc: redefined to use accessibility information	464
\@Gls@: removed \glslabel (defined in \@gls@link)	146	\Genacrfullformat: redefined to use accessibility information	446
\@Gls@entry@field: new	175	\genacrfullformat: redefined to use accessibility information	446
\@Glspl@: removed \glslabel (defined in \@gls@link)	149	\Genplacrfullformat: redefined to use accessibility information	447
\@acrlong: removed \glslabel (defined in \@gls@link)	448	\genplacrfullformat: redefined to use accessibility information	447
\@acrshort: removed \glslabel (defined in \@gls@link)	447	\glossentryname: added	
\@gls@: removed \glslabel (defined in \@gls@link)	145	\glsdetoklabel	258
\@gls@access@display: new	438	\gls@defglossaryentry: added	
\@gls@entry@field: new	174	\glsdetoklabel	103
\@gls@fetchfield: new	95		
\@gls@field@link: new	151		
\@gls@link: added			
\glsdetoklabel	132		

replaced #1 with		
\@glo@label	104	
replaced \ifthenelse with		
\ifdefequal	105	
\glsadd: added \glsdetoklabel	186	
\glsaddkey: switched to using		
\@gls@field@link	97	
\glsdetoklabel: new	73	
\glsdisplaynumberlist: added		
\glsdetoklabel	184	
\glsdoifexistsorwarn: new ..	74	
\glsentryaccess: switched to		
using \@gls@entry@field ..	434	
\glsentrydescaccess: switched		
to using \@gls@entry@field	435	
\glsentrydescpluralaccess:		
switched to using		
\@gls@entry@field	435	
\glsentryfirstaccess: switched		
to using \@gls@entry@field	434	
\glsentryfirstplural: added		
\glsdetoklabel	179	
\glsentrylongaccess: switched		
to using \@gls@entry@field	435	
\glsentrylongpluralaccess:		
switched to using		
\@gls@entry@field	436	
\glsentrypluralaccess:		
switched to using		
\@gls@entry@field	435	
\glsentryshortaccess: switched		
to using \@gls@entry@field	435	
\glsentryshortpluralaccess:		
switched to using		
\@gls@entry@field	435	
\glsentrysymbolaccess:		
switched to using		
\@gls@entry@field	435	
\glsentrysymbolpluralaccess:		
switched to using		
\@gls@entry@field	435	
\glsentrytextaccess: switched		
to using \@gls@entry@field	434	
\glsgenacfmt: redefined to use		
accessibility information ..	444	
\glsgenentryfmt: redefined to		
use accessibility information	441	
\glshyperlink: added		
\glsdetoklabel	185	
\glslocalreset: added		
\glsdetoklabel	114	
\glslocalunset: added		
\glsdetoklabel	114	
\glsmoveentry: added		
\glsdetoklabel	109	
replaced \ifthenelse with		
\ifdefequal	109	
\glsrefentry: added		
\glsdetoklabel	256	
\glsreset: added		
\glsdetoklabel	113	
\glsseelist: added		
\expandafter commands ..	222	
\glsstepentry: added		
\glsdetoklabel	256	
\glsstepsubentry: added		
\glsdetoklabel	256	
\glsunset: added		
\glsdetoklabel	114	
\ifglshasdescsuppressed: added		
\glsdetoklabel	75	
fixed typo	75	
\ifglshasentryexists: added		
\glsdetoklabel	73	
\ifglshaschildren: added		
\glsdetoklabel	74	
\ifglshasdesc: added		
\glsdetoklabel	75	
\ifglshasfield: new	77	
\ifglshaslong: added		
\glsdetoklabel	76	
\ifglshasparent: added		
\glsdetoklabel	75	
\ifglshasshort: added		
\glsdetoklabel	76	
\ifglshassymbol: added		
\glsdetoklabel	76	
replaced \ifcempty with		
\ifdefempty and replaced		
\ifx with \ifdefequal	76	
\ifglshasused: added		
\glsdetoklabel	73	
long-sc-short-desc: redefined		
to use accessibility		
information	457	
long-short: redefined to use		
accessibility information ..	456	
long-short-desc: redefined to		
use accessibility information	457	

long-sm-short-desc: redefined to use accessibility information	458	\showgloplural: added \glsdetoklabel	293
\renewacronymstyle: new	277	\showglopluralaccess: added \glsdetoklabel	466
sc-short-long-desc: redefined to use accessibility information	458	\showgloshort: added \glsdetoklabel	296
short-long: commented spurious EOL	279	\showgloshortaccess: added \glsdetoklabel	467
redefined to use accessibility information	456	\showgloshortpluralaccess: added \glsdetoklabel . . .	467
short-long-desc: redefined to use accessibility information	458	\showglosort: added \glsdetoklabel	296
\showglocounter: added \glsdetoklabel	294	\showglosymbol: added \glsdetoklabel	296
\showglodesc: added \glsdetoklabel	295	\showglosymbolaccess: added \glsdetoklabel	466
\showglodescaccess: added \glsdetoklabel	466	\showglosymbolplural: added \glsdetoklabel	296
\showglodescplural: added \glsdetoklabel	296	\showglosymbolpluralaccess: added \glsdetoklabel . . .	466
\showglodescpluralaccess: added \glsdetoklabel . . .	466	\showglotext: added \glsdetoklabel	293
\showglofirst: added \glsdetoklabel	293	\showglotextaccess: added \glsdetoklabel	466
\showglofirstaccess: added \glsdetoklabel	466	\showglotype: added \glsdetoklabel	294
\showglofirsttpl: added \glsdetoklabel	294	\showglouseri: added \glsdetoklabel	294
\showglofirstpluralaccess: added \glsdetoklabel . . .	466	\showglouserii: added \glsdetoklabel	294
\showgloflag: added \glsdetoklabel	297	\showglouseriii: added \glsdetoklabel	295
\showgloindex: added \glsdetoklabel	297	\showglouseriv: added \glsdetoklabel	295
\showglolevel: added \glsdetoklabel	293	\showglouserv: added \glsdetoklabel	295
\showglolong: added \glsdetoklabel	297	\showglouservi: added \glsdetoklabel	295
\showglolongaccess: added \glsdetoklabel	467	sm-short-long-desc: redefined to use accessibility information	459
\showglolongpluralaccess: added \glsdetoklabel . . .	467	4.04 – 2014-03-04 \@gls@getcounterprefix: added warning if no prefix can be formed	221
\showgloname: added \glsdetoklabel	295	4.04 – 2014-03-06 General: added sort key to printgloss group	255
\showglonameaccess: added \glsdetoklabel	466	\@@gls@noidx@nosanitizesort: new	27
\showgloparent: added \glsdetoklabel	293		

<code>\@gls@noidx@sanitizesort:</code>		<code>\gls@defglossaryentry:</code>	
new	27	replaced <code>\ifx</code> with	
<code>\@gls@nosanitizesort: new</code> ..	27	<code>\ifdefvoid</code>	108
<code>\@gls@sanitizesort: new</code> ...	27	<code>\glsnoidxdisplayloc: new</code> ...	250
<code>\@glo@addchildren: new</code>	237	<code>\glsnoidxdisplaylocclishandler:</code>	
<code>\@glo@do@sortentries: new</code> ..	237	new	250
<code>\@glo@grabfirst: new</code>	247	<code>\glsnoidxloclist: new</code>	249
<code>\@glo@sortedinsert: new</code>	238	<code>\glsnoidxlocclishandler: new</code>	249
<code>\@glo@sortentries: new</code>	236	<code>\glsnoidxstripaccents: new</code> ..	28
<code>\@glo@sorthandler@case: new</code> .	239	<code>\makeglossaries: Moved</code>	
<code>\@glo@sorthandler@letter:</code>		definition of <code>\glswrite</code> to	
new	238	<code>\makeglossaries</code>	203
<code>\@glo@sorthandler@nocase:</code>		<code>\makenoidxglossaries: new</code> ..	205
new	239	<code>\printglossary: changed to use</code>	
<code>\@glo@sorthandler@word: new</code> .	238	new <code>\@printglossary</code>	224
<code>\@glo@sortmacro@case: new</code> ..	240	<code>\printnoidxglossaries: new</code> ..	225
<code>\@glo@sortmacro@def: new</code> ...	241	<code>\printnoidxglossary: new</code> ...	225
<code>\@glo@sortmacro@def@do: new</code> .	241	<code>\showgloloclist: new</code>	297
<code>\@glo@sortmacro@letter: new</code> .	240	<code>\warn@noprintglossary:</code>	
<code>\@glo@sortmacro@nocase: new</code> .	241	Activate warning in	
<code>\@glo@sortmacro@standard:</code>		<code>\makeglossaries</code>	223
new	240	<code>\writeist: checked for definition</code>	
<code>\@glo@sortmacro@use: new</code> ...	242	of <code>\glswrite</code>	189, 193
<code>\@glo@sortmacro@word: new</code> ..	239	4.06 – 2014-03-12	
<code>\@gls@noidx@do: new</code>	248	General: added	
<code>\@gls@noidx@getgrouptitle:</code>		<code>\glsifhyper</code>	167–174
new	262	<code>\@GLS@: added \glsifhyper</code> ...	147
<code>\@gls@noref@warn: new</code>	209	<code>\@GLSpl: added \glsifhyper</code> ..	150
<code>\@gls@reference: new</code>	250	<code>\@Gls@: added \glsifhyper</code> ...	146
<code>\@gls@warnonglossdefined: new</code>	25	<code>\@Glspl@: added \glsifhyper</code> ..	149
<code>\@gls@warnonthe glossdefined:</code>		<code>\@gls@: added \glsifhyper</code> ...	146
new	25	<code>\@gls@numbersdef: added hook</code>	
<code>\@no@makeglossaries: new</code> ...	209	to set toc title	45
<code>\@print@glossary: new</code>	227	<code>\@gls@symbolsdef: added hook</code>	
<code>\@print@noidx@glossary: new</code> .	242	to set toc title	44
<code>\@print@gloss@setsort: new</code> ..	225	<code>\@glsdisp: added \glsifhyper</code>	150
<code>\@printglossary: new</code>	225	<code>\@glspl@: added \glsifhyper</code> ..	148
<code>alttree: moved hangindent and</code>		acronym: added hook to set toc	
<code>parindent</code> assignments outside		title	22
level test	417	acronyms: added hook to set toc	
<code>\compatibleglossentry: changed</code>		title	22
<code>\newcommand</code> to <code>\def</code> as is		<code>\glsdefmain: added hook to set</code>	
may or may not be defined	429	toc title	21
<code>\compatiblesubglossentry:</code>		4.07 – 2014-04-04	
changed <code>\newcommand</code> to <code>\def</code>		<code>\@glossarysection: added</code>	
as is may or may not be		optional argument when using	
defined	429	unstarred version	60
<code>\glo@grabfirst: new</code>	247	<code>\@gls@noidx@do: added \global</code>	
		in case it's used in a	
		tabular-like style	248

<code>\Acrfullplfmt</code> : fixed no case		removed <code>\@GLSsymbol</code>	159
change bug	272	removed <code>\@GLssymbol</code>	159
<code>\glsletentryfield</code> : new	174	removed <code>\@sglssymbol</code>	158
4.08 – 2014-07-30		removed <code>\@GLStext</code>	152
General: added entrycounter		removed <code>\@sGlstext</code>	152
option to <code>printgloss</code> family	254	removed <code>\@sglstext</code>	151
added nopostdot option to		removed <code>\@GLSuseriii</code>	163
<code>printgloss</code> family	254	removed <code>\@Glsuseriii</code>	163
added subentrycounter option		removed <code>\@sglsuseriii</code>	162
to <code>printgloss</code> family	254	removed <code>\@GLSuserii</code>	162
explicitly initialise hyper key	128	removed <code>\@Glsuserii</code>	162
moved <code>\glsifhyper</code>	167–174	removed <code>\@sglsuserii</code>	161
removed <code>\@sACRlongpl</code>	173	removed <code>\@GLSuseriv</code>	164
removed <code>\@sAcrlongpl</code>	173	removed <code>\@Glsuseriv</code>	164
removed <code>\@sacrlongpl</code>	172	removed <code>\@sglsuseriv</code>	163
removed <code>\@sACRlong</code>	171	removed <code>\@GLSuseri</code>	161
removed <code>\@sAcrlong</code>	171	removed <code>\@sGlsuseri</code>	161
removed <code>\@sacrlong</code>	170	removed <code>\@sglsuseri</code>	160
removed <code>\@sACRshortpl</code>	169	removed <code>\@GLSuservi</code>	166
removed <code>\@sAcrshortpl</code>	169	removed <code>\@Glsuservi</code>	166
removed <code>\@sacrshortpl</code>	168	removed <code>\@sglsuservi</code>	165
removed <code>\@sACRshort</code>	167	removed <code>\@GLSuserv</code>	165
removed <code>\@sAcrshort</code>	167	removed <code>\@sGlsuserv</code>	165
removed <code>\@sacrshort</code>	166	removed <code>\@sglsuserv</code>	164
removed <code>\@sgls@link</code>	130	removed <code>\@GLS</code>	147
removed <code>\@GLSdescplural</code>	158	removed <code>\@Gls</code>	146
removed <code>\@GLSdescplural</code>	158	removed <code>\@sgls</code>	145
removed <code>\@sglsdescplural</code>	157	removed <code>\@thirdofthree</code>	
removed <code>\@GLSdesc</code>	157	(defined in kernel)	145
removed <code>\@Glsdesc</code>	157	removed <code>sPGLS</code>	427
removed <code>\@sglsdesc</code>	156	removed <code>sPgls</code>	425
removed <code>\@sglsdisp</code>	150	removed <code>spgls</code>	424
removed <code>\@GLSfirstplural</code>	155	removed <code>sPGLSpl</code>	427
removed <code>\@Glsfirstplural</code>	155	removed <code>sPglspl</code>	426
removed <code>\@sglsfirstplural</code>	154	removed <code>spglspl</code>	424
removed <code>\@GLSfirst</code>	153	<code>\@ACRlong</code> : added	
removed <code>\@Glsfirst</code>	153	<code>\do@gl@link@checkfirsthyper</code>	
removed <code>\@sglsfirst</code>	152	449
removed <code>\@GLSname</code>	156	<code>\@ACRshort</code> : added	
removed <code>\@Glsname</code>	156	<code>\do@gl@link@checkfirsthyper</code>	
removed <code>\@sglsname</code>	155	448
removed <code>\@GLSplural</code>	154	<code>\@Acrlong</code> : added	
removed <code>\@Glsplural</code>	154	<code>\do@gl@link@checkfirsthyper</code>	
removed <code>\@sglsplural</code>	153	448
removed <code>\@GLSpl</code>	149	<code>\@Acrshort</code> : added	
removed <code>\@Glspl</code>	148	<code>\do@gl@link@checkfirsthyper</code>	
removed <code>\@sglspl</code>	148	447
removed <code>\@GLSsymbolplural</code>	160	<code>\@GLS@</code> : moved <code>\glsifhyper</code>	147
removed <code>\@GLssymbolplural</code>	160	moved check for first use to	
removed <code>\@sglssymbolplural</code>	159	<code>\@gl@link</code>	147

\@GLSpl: moved \glsifhyper ..	150	\ACRfullpl: removed	
moved check for first use to		\s@ACRfullpl	273
\@gls@link	150	switched to using	
\@Gls@: moved \glsifhyper ...	146	\@gls@hyp@opt	273
moved check for first use to		\Acrfullpl: removed	
\@gls@link	146	\s@Acrfullpl	273
\@Glspl@: moved \glsifhyper .	149	switched to using	
moved check for first use to		\@gls@hyp@opt	273
\@gls@link	149	\acrfullpl: removed	
\@acrlong: added		\s@acrfullpl	273
\do@gls@link@checkfirsthyper		switched to using	
.....	448	\@gls@hyp@opt	273
\@acrshort: added		\ACRlong: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	171
.....	447	\Acrlong: switched to using	
\@closegls: new	200	\@gls@hyp@opt	171
\@gls@: moved \glsifhyper ...	146	\acrlong: switched to using	
moved check for first use to		\@gls@hyp@opt	170
\@gls@link	146	\ACRlongpl: switched to using	
\@gls@automake: new	200	\@gls@hyp@opt	173
\@gls@doautomake: new	41	\Acrlongpl: switched to using	
\@gls@field@link: added		\@gls@hyp@opt	173
assignment of		\acrlongpl: switched to using	
\do@gls@link@checkfirsthyper		\@gls@hyp@opt	172
.....	151	\ACRshort: switched to using	
\@gls@forbidtextext: new	79	\@gls@hyp@opt	167
\@gls@hyp@opt: new	129	\Acrshort: switched to using	
\@gls@link: removed		\@gls@hyp@opt	167
redundancy	132	\acrshort: switched to using	
renamed \gls@type to		\@gls@hyp@opt	166
\glstype	132	\ACRshortpl: switched to using	
\@gls@link@checkfirsthyper:		\@gls@hyp@opt	169
new	131	\Acrshortpl: switched to using	
\@glsdisp: moved \glsifhyper	150	\@gls@hyp@opt	169
moved check for first use to		\acrshortpl: switched to using	
\@gls@link	150	\@gls@hyp@opt	168
\@glspl@: moved \glsifhyper .	148	altlongragged4col: fixed bug	
moved check for first use to		that displayed description	
\@gls@link	148	instead of symbol	325
\@ignored@glossaries: new ..	83	automake: new	40
\ACRfull: removed \s@ACRfull	273	\forallacronyms: new	71
switched to using		\GLS: switched to using	
\@gls@hyp@opt	272	\@gls@hyp@opt	147
\Acrfull: removed \s@Acrfull	272	\Gls: switched to using	
switched to using		\@gls@hyp@opt	146
\@gls@hyp@opt	272	\gls: switched to using	
\acrfull: removed \s@acrfull	272	\@gls@hyp@opt	145
switched to using		\gls@defglossaryentry: added	
\@gls@hyp@opt	272	check for ignored glossary .	105
		\gls@istfilebase: new	55

\glsaddkey: removed		
\@sGLS@user@<key>	98	
removed \@sGLS@user@<key>	98	
removed \@sgls@user@<key>	97	
switched to using		
\@gls@hyp@opt	97, 98	
\GLSdesc: switched to using		
\@gls@hyp@opt	157	
\Glsdesc: switched to using		
\@gls@hyp@opt	156	
\glsdesc: switched to using		
\@gls@hyp@opt	156	
\GLSdescplural: switched to		
using \@gls@hyp@opt	158	
\Glsdescplural: switched to		
using \@gls@hyp@opt	157	
\glsdescplural: switched to		
using \@gls@hyp@opt	157	
\glsdisablehyper: added		
\KV@glslink@hyperfalse to		
definition	144	
\glsdisp: switched to using		
\@gls@hyp@opt	150	
\glsdohyperlink: new	144	
\glsdohypertarget: new	142	
\glsenablehyper: added		
\KV@glslink@hypertrue to		
definition	145	
\GLSfirst: switched to using		
\@gls@hyp@opt	153	
\Glsfirst: switched to using		
\@gls@hyp@opt	152	
\glsfirst: switched to using		
\@gls@hyp@opt	152	
\GLSfirstplural: switched to		
using \@gls@hyp@opt	155	
\Glsfirstplural: switched to		
using \@gls@hyp@opt	154	
\glsfirstplural: switched to		
using \@gls@hyp@opt	154	
\glsifhyper: deprecated	129	
\glslink: switched to using		
\@gls@hyp@opt	130	
\glslinkcheckfirsthyperhook:		
new	131	
\glslinkvar: new	129	
\GLSname: switched to using		
\@gls@hyp@opt	156	
\Glsname: switched to using		
\@gls@hyp@opt	155	
\glsname: switched to using		
\@gls@hyp@opt	155	
\glsname: switched to using		
\@gls@hyp@opt	155	
\GLSpl: switched to using		
\@gls@hyp@opt	149	
\Glspl: switched to using		
\@gls@hyp@opt	148	
\glspl: switched to using		
\@gls@hyp@opt	147	
\GLSplural: switched to using		
\@gls@hyp@opt	154	
\Glsplural: switched to using		
\@gls@hyp@opt	153	
\glsplural: switched to using		
\@gls@hyp@opt	153	
\glsspace: new	271	
\GLSsymbol: switched to using		
\@gls@hyp@opt	159	
\Glsymbol: switched to using		
\@gls@hyp@opt	158	
\glssymbol: switched to using		
\@gls@hyp@opt	158	
\GLSsymbolplural: switched to		
using \@gls@hyp@opt	160	
\Glsymbolplural: switched to		
using \@gls@hyp@opt	159	
\glssymbolplural: switched to		
using \@gls@hyp@opt	159	
\GLStext: switched to using		
\@gls@hyp@opt	151	
\Glstext: switched to using		
\@gls@hyp@opt	152	
\glstext: switched to using		
\@gls@hyp@opt	151	
\glstreenamefmt: new	410	
\GLSuseri: switched to using		
\@gls@hyp@opt	161	
\Glsuseri: switched to using		
\@gls@hyp@opt	160	
\glsuseri: switched to using		
\@gls@hyp@opt	160	
\GLSuserii: switched to using		
\@gls@hyp@opt	162	
\Glsuserii: switched to using		
\@gls@hyp@opt	161	
\glssuserii: switched to using		
\@gls@hyp@opt	161	
\GLSuseriii: switched to using		
\@gls@hyp@opt	163	
\Glsuseriii: switched to using		
\@gls@hyp@opt	162	

\glsuseriii: switched to using \@gls@hyp@opt	162	\@gls@glossary: Renamed \@glossary to \@gls@glossary	212
\GLSuseriv: switched to using \@gls@hyp@opt	164	alttree: moved space after symbol	417, 418
\Glsuseriv: switched to using \@gls@hyp@opt	163	\glspercentchar: new	188
\glsuseriv: switched to using \@gls@hyp@opt	163	\glstildechar: new	188
\GLSuseriv: switched to using \@gls@hyp@opt	165	4.11 – 2014-09-01 \@do@esc@wrglossary: added hook	219
\Glsuseriv: switched to using \@gls@hyp@opt	164	\gls@wrglossary: renamed from \wrglossary to \gls@wrglossary	213
\glsuseriv: switched to using \@gls@hyp@opt	164	\glsaddprotectedpagefmt: new	215
\GLSuseriv: switched to using \@gls@hyp@opt	166	\glsbackslash: new	187
\Glsuseriv: switched to using \@gls@hyp@opt	165	sanitize: none option	32
\glsuseriv: switched to using \@gls@hyp@opt	165	4.12 – 2014-11-22 \@gls@addpredefinedattributes: Added glsignore attribute ..	65
\ifignoredglossary: new	83	\@gls@adjustmode: new	186
\newglossary: added starred version	80	\@gls@notranslatorhook: removed	32
\newignoredglossary: new ...	82	\@gls@toc: added \protect to \numberline	61
\ns@newglossary: added \@gls@type@{name}@log	81	\@gls@usetranslator: new ...	32
new	80	\glsacrpluralsuffix: new ...	51
\p@gls@hyp@opt: new	130	\glsadd: added check for vertical mode	186
\PGLS: changed to use \@gls@hyp@opt	427	\glsaddallunused: replaced @gobble with glsignore ...	187
\Pgls: changed to use \@gls@hyp@opt	425	\glsifusedtranslatordict: new	32
\pgls: changed to use \@gls@hyp@opt	424	\glsglignore: new	187
\PGLSpl: changed to use \@gls@hyp@opt	427	\glsupacrpluralsuffix: new .	52
\Pglspl: changed to use \@gls@hyp@opt	426	\ProvidesGlossariesLang: new	52
\pglspl: changed to use \@gls@hyp@opt	424	\RequireGlossariesLang: new .	52
\s@gls@hyp@opt: new	129	4.13 – 2015-02-03 \indexspace: new ...	306, 329, 410
\s@newglossary: new	80	4.14 – 2015-02-28 \@gls@localreset: new	115
4.09 – 2014-08-12 \glsaddkey: fixed bug in user commands	97	\@gls@localunset: new	114
4.10 – 2014-08-27 \@Gls@acrentryname: new	176	\@gls@reset: new	115
\@Gls@entryname: new	176	\@gls@unset: new	115
		\@@newglossaryentry@defcounters: new	116
		\@cGls: new	120
		\@cGls@: new	120
		\@cGlspl@: new	121
		\@cgls: new	119
		\@cgls@: new	119
		\@cglspl: new	120, 121

\cglspl@: new	120	\@GLSpl: added	
\@gls@entry@count: new	119	\glspostlinkhook	150
\@gls@increment@currcount:		\@Gls@: added	
new	118	\glspostlinkhook	147
\@gls@local@increment@currcount:		\@GLspl@: added	
new	118	\glspostlinkhook	149
\@gls@write@entrycounts: new	119	\@acrlong: added	
\@glslocalreset: new	115	\glspostlinkhook	448
\@glslocalunset: new	114	\@acrshort: added	
\@glsreset: new	115	\glspostlinkhook	447
\@glsunset: new	114	\@gls@: added	
\@newglossaryentry@defcounters:		\glspostlinkhook	146
new	109	\@gls@@link: added	
\cGls: new	120	\glspostlinkhook	130
\cgls: new	119	\@gls@field@link: added	
\cGlsformat: new	120	\glspostlinkhook	151
\cglsformat: new	119	\@gls@link: moved definition of	
\cGlspl: new	121	\glsifhyperon outside of this	
\cglspl: new	120	macro	132
\cGlsplformat: new	121	\@glsdisp: added	
\cglsplformat: new	121	\glspostlinkhook	150
\gls@defdocnewglossaryentry:		\@GLspl@: added	
new	91	\glspostlinkhook	148
\glsenableentrycount: new	116	\glsacspace: new	278
\glslocalreset: switched to		\glsadd: changed	
\@glslocalreset	114	\@do@wrglossary to	
\glslocalunset: switched to		\@do@wrglossary	186
\@glslocalunset	114	\glsfielddef: new	100
\glsreset: switched to		\glsfieldedef: new	99
\@glsreset	113	\glsfieldfetch: new	100
\glsunset: switched to		\glsfieldgdef: new	100
\@glsunset	114	\glsfieldxdef: new	99
4.15 – 2015-03-16		\glsifhyperon: moved definition	
General: bug fix replaced		of \glsifhyperon	131
\@glo@type with \glstype	173	\glslinkpostsetkeys: new	131
4.16 – 2015-06-18		\glspostlinkhook: new	131
\glsaddstoragekey: new	96	\glswriteentry: new	214
4.16 – 2015-07-08		\ifglsfieldcseq: new	102
General: added		\ifglsfielddefeq: new	101
\glspostlinkhook	167–174	\ifglsfieldefeq: new	101
\@ACRlong: added		long-sp-short: new	278
\glspostlinkhook	449	long-sp-short-desc: new	280
\@ACRshort: added		\showglofield: new	297
\glspostlinkhook	448		
\@Acrlong: added		4.18 – 2015-09-09	
\glspostlinkhook	449	General: split mfirstuc into	
\@Acrshort: added		separate bundle	4
\glspostlinkhook	448	4.19 – 2015-10-31	
\@GLS@: added		\glstreenamebox: new	416
\glspostlinkhook	147		

4.19 – 2015-11-22			
\@gls@link@nocheckfirsthyper:		\glsentrytitlecase: new	180
new	151	\glsfindwidesttoplevelname:	
\@gls@preglossaryhook: new .	225	new	415
\@printglossary: added		\glslistgroupheaderfmt: new .	306
\@gls@preglossaryhook . .	227	\glslistnavigationitem: new .	306
\do@glsglisablehyperinlist:		\glstreegroupheaderfmt: new .	410
new	131	\glstreenavigationfmt: new .	410
\doifglossarynoexistsordo:		\ifglswrallowprimitivemods:	
new	74	new	217
\gls@gobbleopt: new	80	list: fixed missing space before	
\glsglsoifexistsordo: new	74	description	307
4.20 – 2015-11-30		long: fixed typo in	
\@gls@link: added		\glossentrydesc	311
\@gls@setdefault@glslink@opts		mcolaltnonnamespannav: new	334
.	132	mcolindexspannav: new	330
added \glsglsonohyperlink		mcoltreenonnamespannav: new .	332
when hyperlink is suppressed	132	mcoltreesspannav: new	331
\@gls@setdefault@glslink@opts:		super4col: fixed bug in	
new	131	\glossentry	339
\gls@checkseeallowed@preambleonly:		4.23 – 2016-04-30	
new	86	altlongragged4col: check for	
\glsglsonohyperlink: new	144	nogroupskip changed	325
4.21 – 2016-01-24		altsuperragged4col: check for	
General: changed checkfirsthyper		nogroupskip changed	346
assignment	166–174	\glscurrentfieldvalue: new .	78
\@printglossary: warn if no		\ifglshasfield: added	
style has been set	225	\glscurrentfieldvalue	77, 78
altlong4col-booktabs: new . .	318	long: check for nogroupskip	
altlongragged4col-booktabs:		changed	311
new	319	long-booktabs: check for	
\glsglLtpenaltycheck: new	320	nogroupskip changed	317
\glsglpatchLToutput: new	320	long3col: check for nogroupskip	
\glsglspenaltygroupskip: new . .	320	changed	312
long-booktabs: new	317	long3col-booktabs: check for	
long3col-booktabs: new	317	nogroupskip changed	317
long4col-booktabs: new	318	long4col: check for nogroupskip	
longragged-booktabs: new . . .	319	changed	314
longragged3col-booktabs: new	319	long4col-booktabs: check for	
\setglossarystyle: set default		nogroupskip changed	318
style if not already set	264	longragged: check for	
4.22 – 2016-04-19		nogroupskip changed	322
\@do@esc@wrglossary: added		longragged3col: check for	
check for \@arabic	218	nogroupskip changed	324
added test to allow temporary		super: check for nogroupskip	
primitive modifications and		changed	335
added arabic case	218	super3col: check for nogroupskip	
\gls@arabicpage: new	215	changed	337
\gls@protected@pagefmts:		super4col: check for nogroupskip	
added arabic to list	214	changed	339

superragged: check for nogroupskip changed	343	outside of theglossary environment	246
superragged3col: check for nogroupskip changed	345	\glsnavhyperlinkname: new . . .	300
4.24 – 2016-05-27		4.30 – 2017-06-11	
\@gls@extramakeindexopts: new	198	\@glo@autosee: new	108
\@gls@glossary: added check for debug mode	212	\@glo@autoseehook: new	109
\@gls@see@noindex: new	9	\@glo@check@sortallowed: new	17
debug: new	7	\@gls@noidx@do: letter group assignment made global . . .	249
\glsnomakeindexwarning: new .	61	\@gls@setupsort@def: added check for register	18
\GlsSetQuote: new	195	\@gls@setupsort@none: new . .	20
\GlsSetWriteIstHook: new . . .	195	\@xdycrossrefhook: new	67
seenindex: new	9	\@xdylocationclassorder: bug fix: changed \edef to \def .	68
4.25 – 2016-06-09		\glosortentrieswarning: new .	25
General: changed \DeclareRobustCommand to \newrobustcmd and changed \@ifundefined to \ifcsundef	441	\@gls@set@xr@key: new	85
\@gls@enablesavenonumberlist: new	87	\@gls@xr@key: new	86
\@gls@initnonumberlist: new .	87	\GlsAddXdyLocation: bug fix: changed #1 to #2	67
\@gls@savenonumberlist: new .	87	\glsnoidxstripaccents: added \a	29
4.26 – 2016-10-12		added \TH, \dh and \DH	29
\@glossary@default@style: added check for classicthesis	12	4.31 – 2017-08-10	
\glstreechildpredesc: new . .	411	nolist: added check for “list” style	14
\glstreeitem: new	410	4.31 – 2017-09-10	
\glstreepredesc: new	411	style: changed \renewcommand to \def	12
\glstreesubitem: new	410	4.32 – 2017-08-24	
\glstreesubsubitem: new	411	\@glsnavhypertarget: new . . .	302
mcolindex: replaced \@idxitem with \glstreeitem	329	\@glsshowtarget: new	8
mcolindexspannav: replaced \@idxitem with \glstreeitem	330	\@glsshowtarget: new	8
4.28 – 2017-01-07		4.33 – 2017-09-20	
\glspatchtabularx: new	113	\@do@esc@wrglossary: added \@gls@the and \@gls@number	218
4.29 – 2017-01-19		renamed from \@do@esc@wrglossary . . .	217
General: added check for \@glxtr@doaccsupp	428	\@do@noesc@wrglossary: new .	216
\@gls@noidx@do: current letter group assignment made global	249	\@do@wrglossary: changed to check for esclocations	216
\@gls@old@print@noidx@glossary: moved definition of \@gls@currentlettergroup		\@gls@missinglang@warn: new .	25
		esclocations: new	13
		\GlsSetXdyFirstLetterAfterDigits: added starred version	188
		\GlsSetXdyNumberGroupOrder: new	188

4.34 – 2017-11-03	
\glsshowtarget: modified to check for math mode and inner	8
mcolalttreespannav: removed spurious space	334
4.35 – 2017-11-14	
\glsadd: added \@gls@setsort (in case of sort=use)	186
4.36 – 2018-03-07	
\@gls@glossary: removed \index	213
4.37 – 2018-04-07	
\gls@begindocdefs: new	93
4.38 – 2018-05-10	
\@gls@define@glossaryentrycounter: added check for existence of glossaryentry counter	15
new	15
\@gls@define@glossarysubentrycounter: new	16
prepended \currentglossary. to \theHglossarysubentry and removed spurious eol space	16
\glsaccsupp: added braces around actual text argument	438
\glsentrycounterlabel: bug fix: move conditional inside command	257
\GlsEntryCounterLabelPrefix: new	254
\glsentryitem: bug fix: move conditional inside command	257
\glsrefentry: bug fix: move conditional inside command	256
\glsresetentrycounter: bug fix: move conditional inside command	256
\glsresetsubentrycounter: bug fix: move conditional inside command	256
\glsstepentry: bug fix: move conditional inside command	256
\glsstepsubentry: bug fix: move conditional inside command	256
\glssubentrycounterlabel: bug fix: move conditional inside command	257
\glssubentryitem: bug fix: move conditional inside command	257
\showglongnameaccess: bug fix: corrected field (was showing text access field)	466
4.40 – 2018-06-01	
\istfile: changed \def to \providecommand	211
\makenoidxglossaries: false . .	206
4.41 – 2018-07-23	
General: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr	253
\@gls@override@glossary: new	43
debug: changed \val and \nr to \gls@debug@val and \gls@debug@nr	7
\glossary: added warning	43
\gls@original@glossary: new . .	43
\gls@original@makeglossary: new	43
kernelglossredefs: new	43
\makeglossaries: removed redefinition of \makeglossary	204
\makeglossary: added warning	43
nonumberlist: changed \val and \nr to \gls@nonumberlist@val and \gls@nonumberlist@nr	86
numberedsection: changed \val and \nr to \gls@numberedsection@val and \gls@numberedsection@nr . .	11
seenoindex: changed \val and \nr to \gls@seenoindex@val and \gls@seenoindex@nr . . .	9
translate: changed \val and \nr to \gls@translate@val and \gls@translate@nr	33
4.42 – 2019-01-06	
\@gls@@automake@immediate: new	202
\@gls@automake@immediate: new	201
altlong4col-booktabs: removed superfluous \glspatchLToutput	318

automake: switch from boolean to choice	40	\@GLSuserv@: added accessibility support	454
\gls@automake@nr: new	40	\@GLSuservi@: added accessibility support	454
\glsfielddedef: changed from \edef to \protected@csedef	99	\@GLSdesc@: added accessibility support	451
\glsfieldxdef: changed from \edef to \protected@csxdef	99	\@GLSdescplural@: added accessibility support	451
\ifglsautomake: now defined explicitly instead of through boolean key	40	\@GLSfirst@: added accessibility support	450
noglossaryindex: new	45	\@GLSfirstplural@: added accessibility support	450
4.43 – 2019-09-28		\@GLSname@: added accessibility support	451
\glsnoidxstripaccents: add check for LaTeX version 2019/10/01	29	\@GLSplural@: added accessibility support	450
4.44 – 2019-12-06		\@GLSsymbol@: added accessibility support	452
\glsprefix@record@hook: new	423	\@GLSsymbolplural@: added accessibility support	452
4.45 – 2020-02-13		\@GLStext@: added accessibility support	449
General: removed backward compatibility use of symbol key	432	\@GLSuseri@: added accessibility support	452
\@do@write@glslabels: new	38	\@GLSuserii@: added accessibility support	453
\@glsshowtarget: new	9	\@GLSuseriii@: added accessibility support	453
\@GLSdesc@: added accessibility support	451	\@GLSuseriv@: added accessibility support	453
\@GLSdescplural@: added accessibility support	451	\@GLSuserv@: added accessibility support	454
\@GLSfirst@: added accessibility support	450	\@GLSuservi@: added accessibility support	454
\@GLSfirstplural@: added accessibility support	450	\@do@write@glslabels: new	38
\@GLSname@: added accessibility support	451	\@domakeglossaries: new	37
\@GLSplural@: added accessibility support	450	\@gls@access@display: deprecated	438
\@GLSsymbol@: added accessibility support	452	\@gls@fieldaccess@display: new	438
\@GLSsymbolplural@: added accessibility support	452	\@glsdesc@: added accessibility support	451
\@GLStext@: added accessibility support	449	\@glsdescplural@: added accessibility support	451
\@GLSuseri@: added accessibility support	452	\@glsfirst@: added accessibility support	450
\@GLSuserii@: added accessibility support	453	\@glsfirstplural@: added accessibility support	450
\@GLSuseriii@: added accessibility support	453	\@glsname@: added accessibility support	451
\@GLSuseriv@: added accessibility support	453		

\@glsplural@: added accessibility support	450	\Glsentryprefixfirstplural: added \glsdetoklabel	422
\@glsshowaccsupp: new	9	\glentryprefixfirstplural: added \glsdetoklabel	422
\@glssymbol@: added accessibility support	452	\Glsentryprefixplural: added \glsdetoklabel	423
\@glssymbolplural@: added accessibility support	452	\glentryprefixplural: added \glsdetoklabel	422
\@glstext@: added accessibility support	449	\glentrytitlecase: added existence check	180
\@glseri@: added accessibility support	452	\glentryuseriaccess: new	436
\@glserii@: added accessibility support	453	\glentryuseriiaaccess: new	436
\@glseriii@: added accessibility support	453	\glentryuseriiiaaccess: new	436
\@glseriv@: added accessibility support	453	\glentryuserivaccess: new	436
\@glservi@: added accessibility support	454	\glentryuserviaccess: new	436
\@org@Gls@acentryname: added check for \glsshortaccessdisplay	177	\glsfieldaccsupp: new	437
acronyms: changed \renewcommand to \def	22	\glsfirstaccessdisplay: added check for existence	439
debug: showaccsupp	7	\glsfirstpluralaccessdisplay: added check for existence	439
disablemakegloss: new	37	\glslongaccessdisplay: added check for existence	440
\gls@accessibility: new	429	\glslongpluralaccessdisplay: added check for existence	440
\gls@accsupp@engine: new	429	\glsnameaccessdisplay: added check for existence	438
\glsaccessibility: new	429	\glspluralaccessdisplay: added check for existence	439
\glsdefaultshortaccess: new	465	\glsprefixsep: new	424
\glsdescriptionaccessdisplay: added check for existence	439	\glssee: switched to \newrobustcmd	222
\glsdescriptionpluralaccessdisplay: added check for existence	439	\glsseeformat: switched to \newrobustcmd	222
\glentrydescpluralaccess: corrected field reference	435	\glsseeitem: switched to \newrobustcmd	223
\glentryfirstpluralaccess: switched to using \@gls@entry@field	435	\glsseelist: switched to \newrobustcmd	222
\glentryparent: new	180	\glsshortaccessdisplay: added check for existence	440
\Glsentryprefix: added \glsdetoklabel	423	\glsshortaccsupp: new	437
\glentryprefix: added \glsdetoklabel	422	\glsshortplaccsupp: new	437
\Glsentryprefixfirst: added \glsdetoklabel	422	\glsshortpluralaccessdisplay: added check for existence	440
\glentryprefixfirst: added \glsdetoklabel	422	\glsshowaccsupp: new	9
		\glsshowtargetfont: new	8
		\glsshowtargetouter: new	8
		\glsshowtargetsymbols: new	8
		\glssymbolaccessdisplay: added check for existence	439

<code>\glssymbolpluralaccessdisplay:</code>		<code>\ifglossaryexists:</code> added	
added check for existence	439	starred form	71
<code>\glstextaccessdisplay:</code> added		<code>\s@ifglossaryexists:</code> new	72
check for existence	438	<code>\setglossarypreamble:</code> switched	
<code>\glsuseriaccessdisplay:</code> new	440	to starred form of	
<code>\glsuseriiaccessdisplay:</code> new	440	<code>\ifglossaryexists</code>	58
<code>\glsuseriiiaccessdisplay:</code>		4.47 – 2021-09-20	
new	440	General: changed <code>\x</code> to <code>\@gls@x</code>	267
<code>\glsuserivaccessdisplay:</code> new	441	<code>\@do@esc@wrglossary:</code> changed	
<code>\glsuservaccessdisplay:</code> new	441	<code>\edef</code> to <code>\protected@edef</code>	219
<code>\glsuserviiaccessdisplay:</code> new	441	<code>\@do@noesc@wrglossary:</code>	
<code>\ifglshaschildren:</code> made robust	74	changed <code>\edef</code> to	
<code>\ifglshasfield:</code> made robust	77	<code>\protected@edef</code>	216
<code>\ifglshaslong:</code> made robust	76	<code>\@do@write@glslabels:</code> fully	
<code>\ifglshasprefix:</code> added		expand list	38
<code>\glsdetoklabel</code>	423	<code>\@gls@@default@acr@descplural:</code>	
<code>\ifglshasprefixfirst:</code> added		new	289
<code>\glsdetoklabel</code>	423	<code>\@gls@@default@acr@firstplural:</code>	
<code>\ifglshasprefixfirstplural:</code>		new	289
added <code>\glsdetoklabel</code>	423	<code>\@gls@@default@acr@plural:</code>	
<code>\ifglshasprefixplural:</code> added		new	289
<code>\glsdetoklabel</code>	423	<code>\@gls@noexpand@assign@default@plurals:</code>	
<code>\ifglshasshort:</code> made robust	76	new	289
<code>\ifglshassymbol:</code> made robust	76	<code>\@glo@do@sortentries:</code> changed	
<code>\makeglossaries:</code> let		<code>\edef</code> to <code>\protected@edef</code>	237
<code>\@makeglossary</code> to <code>\@gobble</code>		<code>\@glo@sortentries:</code> changed	
instead of <code>\relax</code>	204	<code>\edef</code> to <code>\protected@edef</code>	236
<code>restormakegloss:</code> new	38	<code>\@glo@storeentry:</code> changed	
<code>user1access:</code> new	431	<code>\edef</code> to <code>\protected@edef</code>	110
<code>user2access:</code> new	431	<code>\@gls@checkesc:</code> new	141
<code>user3access:</code> new	431	<code>\@gls@deprecated@acr@warning:</code>	
<code>user4access:</code> new	431	new	291
<code>user5access:</code> new	432	<code>\@gls@link:</code> changed <code>\edef</code> to	
<code>user6access:</code> new	432	<code>\protected@edef</code>	132
<code>writeglslabels:</code> new	39	<code>\@gls@setupsort@use:</code> changed	
<code>\xglshasfieldaccsupp:</code> new	437	<code>\edef</code> to <code>\protected@edef</code>	19
4.46 – 2020-03-19		<code>\@gls@writedef:</code> added a	
<code>\@ifglossaryexists:</code> new	72	warning if entry has already	
<code>\@printgloss@checkexists:</code>		been defined	94
new	224	<code>\@makeglossaries@warn@noprintglossary:</code>	
<code>\@printgloss@checkexists@allowignored:</code>		new	205
new	224	<code>\@write@glslabelsname:</code> new	39
<code>\@printgloss@checkexists@noignored:</code>		<code>\do@write@glslabelsname:</code> new	39
new	224	<code>\forglsentries:</code> changed <code>\edef</code>	
<code>\@printglossary:</code> replaced		to <code>\protected@edef</code>	71
<code>\ifglossaryexists</code> with		<code>\gls@defdocnewglossaryentry:</code>	
<code>\@printgloss@checkexists</code>	226	removed see warning	91
<code>\doifglossarynoexistsordo:</code>		<code>\gls@defglossaryentry:</code> changed	
switched to starred form of		<code>\edef</code> to	
<code>\ifglossaryexists</code>	74	<code>\protected@edef</code>	103, 104

<code>\gls@docdef@addlabel</code> : new .. 92	<code>\SetDescriptionFootnoteAcronymStyle</code> :
<code>\gls@docdef@labellist</code> : new . 92	added deprecated warnings 290
<code>\gls@glsdefs@addlabel</code> : new . 92	<code>\SetDUASStyle</code> : added deprecated
<code>\gls@glsdefs@labellist</code> : new . 92	warnings 291
<code>\gls@save@numberlist</code> : changed	<code>\SetFootnoteAcronymStyle</code> :
<code>\edef to \protected@edef</code> 223	added deprecated warnings 291
<code>\glsadd</code> : changed <code>\edef to</code>	<code>\SetSmallAcronymStyle</code> : added
<code>\protected@edef</code> 186	deprecated warnings 291
<code>\glsaddall</code> : changed <code>\edef to</code>	<code>writeglslabelnames</code> : new 39
<code>\protected@edef</code> 186	4.48 – 2021-10-19
<code>\glsdefs@newdocentry</code> : new .. 93	<code>\glscapitalisewords</code> : new ... 180
<code>\glsdisplaynumberlist</code> : changed	<code>\glslistexpandedname</code> : new .. 307
<code>\edef to \protected@edef</code> 184	<code>\glslistinit</code> : new 306
<code>\glsfielddef</code> : changed <code>\edef to</code>	<code>\glsunexpandedfieldvalue</code> :
<code>\protected@edef</code> 100	new 101
<code>\glsfielddedef</code> : changed <code>\edef to</code>	4.49 – 2021-11-01
<code>\protected@edef</code> 99	<code>\glslistinit</code> : added
<code>\glsfieldfetch</code> : changed <code>\edef</code>	<code>\glsnavigation</code> 306
<code>to \protected@edef</code> 100	4.50 – 2022-10-14
<code>\glsfieldgdef</code> : changed <code>\edef to</code>	General: added initialisation ... 269
<code>\protected@edef</code> 100	allow markup to be stripped
<code>\glsfieldxdef</code> : changed <code>\edef to</code>	from location 267
<code>\protected@edef</code> 99	corrected description mapping 432
<code>\glsIfListOfAcronyms</code> : changed	don't load glossary-super if xtab
<code>\edef to \protected@edef</code> . 23	has been loaded 14
<code>\glsmoveentry</code> : changed <code>\edef to</code>	No longer requires textcase
<code>\protected@edef</code> 109	with mfirstuc 2.08+ 5
<code>\glsnavhyperlink</code> : changed	removed automatic loading of
<code>\edef to \protected@edef</code> 300	glossaries-compatible-307 ... 46
<code>\GlsSetQuote</code> : added check for	switched to using
backslash 197	<code>\glsuppercase</code>
<code>\glsstepsubentry</code> : changed 152–166, 168, 170, 172, 174
<code>\edef to \protected@edef</code> 256	<code>\@do@esc@wrglossary</code> : moved
<code>\ifglsfieldcseq</code> : changed <code>\edef</code>	<code>\romannumeral</code> inside allow
<code>to \protected@edef</code> 102	primitive mods 218
<code>\ifglshaschildren</code> : changed	<code>\@gls@@default@acr@descplural</code> :
<code>\edef to \protected@edef</code> . 75	removed 289
<code>\ifignoredglossary</code> : changed	<code>\@gls@@default@acr@firstplural</code> :
<code>\edef to \protected@edef</code> . 83	removed 289
<code>\makenoidxglossaries</code> : changed	<code>\@gls@@default@acr@plural</code> :
<code>\edef to \protected@edef</code> 206	removed 289
<code>\new@glossaryentry</code> : allow	<code>\@gls@default@entryfmt</code> :
overwrite if defined in	removed 441
<code>.glsdefs</code> 92	<code>\@gls@navhypertarget</code> : new . 301
<code>\newignoredglossary</code> : changed	<code>\@gls@noexpand@assign@default@plurals</code> :
<code>\edef to \protected@edef</code> . 82	removed 289
<code>\ns@newglossary</code> : changed <code>\edef</code>	<code>\@Gls@acentryname</code> : removed
<code>to \protected@edef</code> 81	mfirstuc internal commands 176
<code>\SetDescriptionDUAAcronymStyle</code> :	<code>\@Glsdisp</code> : new 151
added deprecated warning . 290	<code>\@Glsentryfield</code> : new 175

\@domakeglossaries: added		
check for \nofiles	37	
\@gls@automake: added check for		
void codepage	200	
\@gls@automake@immediate:		
added check for void		
codepage	201	
\@gls@automake@types: new	202	
\@gls@defaultttype: new	79	
\@gls@do@automake@err: new	40	
\@gls@do@glsunset: new	129	
\@gls@entry@field: switched to		
\csuse	174	
\@gls@escbsdq: check		
esclocations setting	135	
\@gls@getcounterprefix:		
changed \edef to \def	220	
\@gls@getgrouptitle: switched		
to L ^A T _E X3 syntax	262	
\@gls@glslink@hyper@update@hook:		
new	128	
\@gls@hypergroup: updated to		
use etoolbox	302	
\@gls@noist@toolate: new	199	
\@gls@removedoption: new	30	
\@gls@requested@languages:		
new	34	
\@gls@restore@glslocal: new	129	
\@gls@sanitizedesc: removed	26	
\@gls@save@glslocal: new	128	
\@gls@setupsort@clear: new	20	
\@gls@defaultplural: Removed	89	
\@org@Gls@acentryname: saved		
original definition of		
\@Gls@acentryname	176	
\@xp@gls@getcounterprefix:		
new	220	
\@xp@gls@hypernumber: new	266	
automake: added makegloss and		
lite options	40	
automakegloss: new	41	
automakeglosslite: new	42	
compatible-2.07: deprecated	42	
compatible-3.07: deprecated	42	
\compatibleglossentry:		
removed	258, 429	
\compatiblesubglossentry:		
removed	260, 429	
\CustomAcronymFields:		
removed	292	
\CustomNewAcronymDef:		
removed	292	
\DefaultNewAcronymDef:		
removed	289, 465	
\defglsdisplay: removed	127	
\defglsdisplayfirst: removed	127	
\DescriptionDUANewAcronymDef:		
removed	290	
\DescriptionFootnoteNewAcronymDef:		
removed	290, 465	
\DescriptionNewAcronymDef:		
removed	290, 465	
dua: switched to using		
\glsupercase	283, 284	
dua-desc: removed \acronymfont		
from around long form	284	
\DUANewAcronymDef: removed	291	
esclocations: default now false	13	
\findrootlanguage: removed	69	
footnote: switched to using		
\glsupercase	285	
\FootnoteNewAcronymDef:		
removed	291, 465	
\glossary: removed	43	
\glossaryentryfield: removed	260	
\glossarymark: switched to using		
\glsupercase	59	
\glossarystyle: removed	265	
\glossarysubentryfield:		
removed	261	
\gls@automake@makegloss: new	40	
\gls@automake@option: new	40	
\gls@codepage: set default		
codepage to utf8 if		
\inputencodingname not		
defined and switched to		
\ifdefvoid	37	
\gls@delimN: renamed \@delimN		
to \gls@delimN	267	
\gls@delimR: renamed \@delimR		
to \gls@delimR	266	
\gls@disablepagerefexpansion:		
added check for esclocations	214	
\gls@noidx@nonumberlist: new	86	
\gls@noidxglossary: added		
\glswrglossdisablelocationcmds		
	209	
\gls@xdy@set@lang: new	37	
\glsaddkey: allow expansion in		
PDF bookmarks	97	

switched to using		
\gluppercase	98	
\GlsAddXdyAttribute: remove		
duplicate counterless		
attribute definitions	63, 64	
\Glsdisp: new	151	
\glsdisplay: removed	127	
\glsdisplayfirst: removed	127	
\glsdoparenifnotempty:		
replaced		
\DeclareRobustCommand with		
\newrobustcmd	291	
\glsdoshowtarget: new	8	
\glsencapwrcontent: new	133	
\Glsentrydesc: allow expansion		
in PDF bookmarks	177	
\Glsentrydescplural: allow		
expansion in PDF		
bookmarks	178	
\Glsentryfirst: allow expansion		
in PDF bookmarks	179	
\Glsentryfirstplural: allow		
expansion in PDF		
bookmarks	180	
\glsentryfmt: obsolete		
commands replaced with		
\glsgenentryfmt	122	
\Glsentrylong: allow expansion		
in PDF bookmarks	183	
\Glsentrylongpl: allow		
expansion in PDF		
bookmarks	183	
\Glsentryname: allow expansion		
in PDF bookmarks	176	
\Glsentryplural: allow		
expansion in PDF		
bookmarks	178	
\Glsentryprefix: allow		
expansion in PDF		
bookmarks	423	
\Glsentryprefixfirst: allow		
expansion in PDF		
bookmarks	422	
\Glsentryprefixfirstplural:		
allow expansion in PDF		
bookmarks	422	
\Glsentryprefixplural: allow		
expansion in PDF		
bookmarks	423	
\Glsentryshort: allow expansion		
in PDF bookmarks	182	
\Glsentryshortpl: allow		
expansion in PDF		
bookmarks	183	
\Glsentrysymbol: allow		
expansion in PDF		
bookmarks	179	
\Glsentrysymbolplural: allow		
expansion in PDF		
bookmarks	179	
\Glsentrytext: allow expansion		
in PDF bookmarks	178	
\glsentrytitlecase: use		
sentence case for PDF		
bookmark	180	
\Glsentryuseri: allow expansion		
in PDF bookmarks	181	
\Glsentryuserii: allow		
expansion in PDF		
bookmarks	181	
\Glsentryuseriii: allow		
expansion in PDF		
bookmarks	181	
\Glsentryuseriv: allow		
expansion in PDF		
bookmarks	182	
\Glsentryuserv: allow expansion		
in PDF bookmarks	182	
\Glsentryuservi: allow		
expansion in PDF		
bookmarks	182	
\glsfielddef: made robust	100	
\glsfieldedef: made robust	99	
\glsfieldfetch: made robust	100	
\glsfieldgdef: made robust	100	
\glsfieldxdef: made robust	99	
\glsgenacfmt: switched to using		
\gluppercase	125	
\glsgenentryfmt: switched to		
using \gluppercase	123	
\glsgetgrouplabel: converted to		
use L ^A T _E X3	263	
\glsgetgrouptitle: made		
robust	262	
\gls glossarymark: switched to		
\gluppercase	59	
\glsifhyper: removed	129	
\glsindexingsetting: new	16	
\glsinlineifhaschildren: new	305	

\Glslink: new	130	\SetCustomDisplayStyle:	
\glslocationcstoencap: new	57	removed	292
\glslowercase: new	4	\SetCustomStyle: removed	292
\glsmfuaddmap: new	5	\SetDefaultAcronymStyle:	
\glsmfublocker: new	5	removed	289
\glsmfuexcl: new	5	\SetDescriptionAcronymDisplayStyle:	
\glsnavhyperlink: changed		removed	290
\@glo@type to		\SetDescriptionAcronymStyle:	
\currentglossary	300	removed	290
\glsnavhypertarget: changed		\SetDescriptionDUAAcronymDisplayStyle:	
\@glo@type to		removed	290
\currentglossary	301	\SetDescriptionDUAAcronymStyle:	
\glsnavigation: changed		removed	290
\@glo@type to		\SetDescriptionFootnoteAcronymDisplayStyle:	
\currentglossary	302	removed	290
\glsnoidxprenumberlist: new	87	\SetDescriptionFootnoteAcronymStyle:	
\glsnumberlistloop: corrected		removed	290
parameter number	209	\SetDUADisplayStyle: removed	291
\glssseeitemformat: added test		\SetDUASStyle: removed	291
for short field	223	\SetFootnoteAcronymDisplayStyle:	
\glssentencecase: new	4	removed	290
\glsshowtargetfonttext: new	8	\SetFootnoteAcronymStyle:	
\glsshowtargetinner: new	8	removed	291
\glstexpdfstring: new	175	\SetGenericNewAcronym:	
\glssupercase: new	5	corrected long plural suffix	275
\GlsWarnAddProtectedPageFmt:		switched to using	
new	215	\glssupercase	275
\glswrglossdisableanchorcmds:		\setglossentrycompatibility:	
new	216	removed	260
\glswrglossdisablelocationcmds:		\SetSmallAcronymDisplayStyle:	
new	214	removed	291
\glswrglosslocationtarget:		\SetSmallAcronymStyle:	
new	216	removed	291
\glswrglosslocationtextfmt:		\SmallNewAcronymDef:	
new	216	removed	291, 465
\ifglsfieldcseq: made robust	102	sort: added clear	17
\ifglsfielddefeq: made robust	101	4.51 – 2022-11-02	
\ifglsfieldeq: made robust	101	General: switch to	
\ifglsfieldvoid: new	77	\TrackIfKnownLanguage	52
\ifglsresetcurrcount: new	116	\@do@wrglossary: added	
\ifglsrallowprimitivemods:		\gls@ifnotmeasuring	214
switch default to false	217	\@gls@@automake@immediate:	
\istfile: removed	211	ensure list is expanded before	
kernelglossredefs: corrected		iterating	202
names of choice variables	43	\@gls@if@measuring: new	111
languages: new	34	alttree: replaced \settowidth	
\makeglossary: removed	43	with \glsmeasurewidth	417, 418
mfirstuc: new	45	\gls@if@measuring: new	111
\SetAcronymStyle: removed	292	\gls@local@stepcounter: new	112
		\gls@org@refstepcounter: new	111

\gls@start@measuring: new ..	112	\Glsentryplural: change map to	
\gls@stop@measuring: new ...	113	exclusion	178
\glsdohypertarget: added		\Glsentryshort: change map to	
measuring and group	142	exclusion	182
\glsfindwidesttoplevelname:		\Glsentryshortpl: change map	
replaced \settowidth with		to exclusion	183
\glsmeasurewidth	415	\Glsentrysymbol: change map to	
\glsifmeasuring: new	112	exclusion	179
\glsmeasuredepth: new	113	\Glsentrysymbolplural: change	
\glsmeasureheight: new	113	map to exclusion	179
\glsmeasurewidth: new	113	\Glsentrytext: change map to	
\glspatchtabularx: switch		exclusion	178
measuring on and off	113	\Glsentryuseri: change map to	
\makeglossaries: ensure list of		exclusion	181
glossaries is expanded before		\Glsentryuserii: change map to	
iterating	205	exclusion	181
4.52 – 2022-11-03		\Glsentryuseriii: change map	
\gls@local@stepcounter:		to exclusion	181
corrected spelling of \@ne ..	112	\Glsentryuseriv: change map to	
4.53 – 2023-09-29		exclusion	182
\@gls@hypergroup: rewritten in		\Glsentryuserv: change map to	
L ^A T _E X3	302	exclusion	182
\glsnavhypergroupdotarget:		\Glsentryuservi: change map to	
new	301	exclusion	182
\glsnavigationitem: new	303	\glslabelhypertarget: new ..	143
4.54 – 2024-04-03		\glslabelhypertargetdefs:	
General: added check for		new	143
\l_datatool_measure_hook_tl		\glslabelhypertargetprefix:	
.....	112	new	143
\gls@printglossary@begin:		\glslabelhypertargetvalue:	
new	227	new	143
\gls@printglossary@end: new ..	227	\glslowercase: ensure argument	
\glsdohyperlinkhook: new ...	144	is expanded	4
\glsdohypertargethook: new ..	143	\glsuppercase: ensure argument	
\Glsentrydesc: change map to		is expanded	5
exclusion	177	4.55 – 2024-11-01	
\Glsentrydescplural: change		\@gls@run@output@dir: new ..	46
map to exclusion	178	\@gls@run@output@file: new ..	46
\Glsentryfirst: change map to		\@gls@run@unrestricted@shell:	
exclusion	179	new	46
\Glsentryfirstplural: change		\@gls@runshell: new	46
map to exclusion	180	\gls@set@xr@key: added check	
\Glsentryfull: change map to		for empty value	85
exclusion	183	locales: new	34
\Glsentrylong: change map to		4.56 – 2025-03-03	
exclusion	183	General: store original sort as	
\Glsentrylongpl: change map to		well as label in actual part	229
exclusion	183	4.57 – 2025-03-12	
\Glsentryname: change map to		\@@gls@noidx@nosanitizesort:	
exclusion	176	check for datatool preprocess	

function	27	tree*: new	348
\@print@noidx@glossary:		4.6 – 2025-04-14	
switched to constructing		mcoltree*: corrected tree header	
token list	242	function name	328
\glsnoidxprecontenthook: new	244	tree*: corrected parameter spec	
4.58 – 2025-03-19		for tree header function	409
\@gls@noidx@nosanitizesort:		corrected sub-group title	350
defer existence check	27	4.9 – 2025-11-09	
\@gls@setupsort@use: internal		\glossarysection: changed to	
count register changed to L3		document command	58
int var	19	section: added option value none	10
switched to L ^A T _E X3	19	5.0 – ??	
mfirstuc: redefine		long-short: use	
\glsmakefirstuc as a long		\glsacrpluralsuffix	
command	45	instead	278
4.59 – 2025-04-11		long-sp-short: use	
\@glo@assign@sortkey:		\glsacrpluralsuffix	
initialise	203	instead	278
\@glo@sortmacro@lettercase:		short-long: use	
new	240	\glsacrpluralsuffix	
\@glo@sortmacro@wordcase:		instead	279
new	239	\showglolongplural: new	297
\@gls@noidx@addtorefs: new	210	\showgloshortplural: new	296
\GLOSSentrydesc: new	259	5.1 – 2026-04-21	
\GlossEntryDesc: new	259	\@@gls@extramakeindexopts:	
\GlossEntryName: new	258	ensure defined	195
\GlossEntrySymbol: new	260	\@glossary@write@seclabel:	
\GlossEntrySymbol: new	260	new	10
\gls@defglossaryentry: moved		\@glossarylabeltype: new	11
auto see after index field has		\@gls@currentglossaryxrlabel:	
been set	108	new	11
\GlsMcolTreeSetup: new	327	\@gls@makeindexapp: new	36
\glsnoidxinithook: new	243	\@gls@provide@glossarygroup:	
\glsnoidxitemhook: new	244	new	264
\GlsNoIdxMissingAction: new	243	\@gls@writeglossarygroup:	
\GlsTreeSetup: new	348	new	263
\GlsTreeStarBox: new	349	\@gls@writeglossarysubgroup:	
\GlsTreeStarItemCounterBox:		new	263
new	350	\@glsmakeindexappname: new	55
\GlsTreeStarNameBox: new	350	\@provide@glossary@seclabel:	
\GlsTreeStarOuterBox: new	350	new	11
\GlsTreeStarSubItemCounterBox:		\GlsAddToMakeIndexOpts: new	195
new	350	\GlsAppToWriteIstHook: new	195
\GlsTreeStarSymbolBox: new	350	\glsSavedGlossaryGroup: new	264
\GlsTreeUpdateWidestNameAndSymbol:		\glswriteglossarygroup: new	39
new	349	\glswriteglossarysubgroup:	
\GlsTreeUpdateWidestNameOrSymbol:		new	39
new	349	saveglossarygroups: new	40
mcoltree*: new	327	v4.8 – 2025-08-11	
style-options: new	12	General: replaced \write18	48

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	
\! 136, 139, 197, 542, 543, 791, 793, 794, 824, 1045, 1047, 1078, 1200, 1202, 1234, 1357, 1359, 1391	\@@glo@sortentries 236, 239–241, 1413, 1415, 1416
\" 29, 136– 139, 141, 479, 540–542, 544, 729, 791–794, 796, 983, 1044– 1047, 1049, 1050, 1138, 1199– 1202, 1204, 1293, 1356–1359, 1361	\@@glo@assign@sortkey 207, <u>255</u> , 578, 596, 830, 848, 1085, 1104, 1241, 1259, 1398, 1425
\# 141, 191, 567, 568, 795, 796, 819, 1049, 1074, 1204, 1229, 1230, 1361, 1386, 1387	\@@glo@list 71, 501, 751, 1007, 1162, 1319
\% 188, 194, 565, 569, 670–672, 817, 821, 923–925, 1071, 1076, 1227, 1231, 1384, 1388	\@@glo@sort 28, 479, 729, 983, 1138, 1293
\& 52, 185, 490, 564, 740, 816, 996, 1070, 1150, 1226, 1307, 1383	\@@glo@type 225, 586, 838, 1094, 1249, 1250, 1406, 1407
\' 29, 479, 729, 983, 1138, 1293	\@@glossarysec 10, <u>10</u> , 58, 60, 402, 471, 494, 720, 721, 744, 975, 1000, 1129, 1154, 1155, 1284, 1311, 1312
\. 15, 29, 472, 479, 722, 729, 976, 983, 1131, 1138, 1285, 1293	\@@glossaryseclabel <u>10</u> , 11, 60, 253, 471, 494, 596, 721, 744, 847, 848, 975, 1000, 1103, 1129, 1130, 1154, 1155, 1259, 1284, 1311, 1312, 1424
\= 29, 479, 729, 983, 1138, 1293	\@@glossarysecstar <u>10</u> , 11, 60, 253, 471, 494, 596, 721, 744, 847, 848, 975, 1000, 1103, 1129, 1130, 1154, 1259, 1284, 1311, 1424
\? 136–138, 197, 540–542, 571, 791– 793, 823, 1044, 1046, 1078, 1199, 1201, 1233, 1356, 1358, 1390	\@@gls@@default@acr@descplural <u>289</u> , 868
\@ 93, 512, 763, 1019, 1173, 1330	\@@gls@@default@acr@firstplural <u>289</u> , 868
\@@delimN 602, 854	\@@gls@@default@acr@plural <u>289</u> , 868
\@@do@@wrglossary 206, 216, 219, <u>219</u> , 577, 583, 584, 829, 834, 835, 1084, 1090, 1091, 1240, 1246, 1247, 1397, 1403	\@@gls@checkactual 140, 544, 795, 1048, 1049, 1203, 1204, 1360, 1361
\@@do@esc@wrglossary 216, <u>217</u> , 582, 583, 834, 835, 1090, 1245, 1246, 1402, 1403	\@@gls@checkbar 139, 543, 794, 1048, 1202, 1203, 1359, 1360
\@@do@noesc@wrglossary . . . 216, <u>216</u> , 582, 834, 1090, 1245, 1402	\@@gls@checkescactual . 138, 541, 542, 792, 793, 1046, 1201, 1358
\@@do@wrglossary 186, 214, <u>216</u> , 564, 581, 582, 816, 833, 834, 1071, 1088, 1090, 1226, 1244, 1245, 1383, 1401, 1402	\@@gls@checkescbar . . . 138, 542, 793, 1047, 1201, 1202, 1358, 1359
\@@do@write@glslabels 38, <u>38</u> , 485, 735, 990, 1144, 1300	\@@gls@checkesclevel . . 139, 542, 543, 793, 794, 1047, 1202, 1359
	\@@gls@checkescquote 137, 198, 541, 572, 573, 792, 824, 825, 1046, 1079, 1200, 1201, 1234, 1235, 1357, 1358, 1391, 1392

<code>\@gls@checklevel</code>	140, 543, 544, 794, 795, 1048, 1203, 1360	<code>\@glslocalreset</code>	115, 115, 117, 525, 527, 776, 777, 1033, 1034, 1188, 1189, 1345, 1346
<code>\@gls@checkquote</code>	137, 196, 540, 541, 571, 791, 792, 822, 823, 1045, 1077, 1200, 1233, 1357, 1390	<code>\@glslocalunset</code>	114, 114, 117, 525, 526, 776, 777, 1033, 1034, 1188, 1189, 1345, 1346
<code>\@gls@codepage</code>	200, 201, 1080–1082, 1236, 1237, 1393, 1394	<code>\@glsreset</code>	115, 115, 117, 525, 526, 776, 777, 1033, 1034, 1188, 1189, 1345, 1346
<code>\@gls@default@entryfmt</code> . .	441, 529, 535, 536, 640, 780, 786, 893	<code>\@glsshowtarget</code>	7, 9, 9, 469, 470, 719, 720, 973, 974, 1128, 1282, 1283
<code>\@gls@expand@field</code> . .	26, 90, 90, 96, 97, 478, 510, 511, 514, 616–619, 621, 623, 625, 627, 661–665, 728, 761, 764, 765, 868, 870, 871, 873, 875, 876, 878, 880, 915–918, 982, 1017, 1020, 1021, 1137, 1171, 1172, 1175, 1176, 1291, 1328, 1329, 1332, 1333	<code>\@glsunset</code>	114, 115, 117, 525, 526, 776, 777, 1033, 1034, 1188, 1189, 1345, 1346
<code>\@gls@extramakeindexopts</code>	195, 195, 200, 202, 203, 570, 576, 822, 828, 1077, 1083, 1232, 1239, 1389, 1395, 1396	<code>\@newglossaryentry@defcounters</code>	116, 116, 526, 777, 1034, 1188, 1189, 1345, 1346
<code>\@gls@fixbraces</code>	221, 222, 585, 837, 1092, 1093, 1248, 1405	<code>\@this@glo@</code>	71, 501, 751, 1007, 1162, 1319
<code>\@gls@navhypertarget</code>	301, 301, 1125	<code>\@ACRfull</code>	273, 604, 856, 1111, 1266, 1267, 1432
<code>\@gls@noexpand@assign@default@plurals</code>	289, 868, 915	<code>\@ACRfullpl</code>	273, 274, 604, 605, 856, 1111, 1267, 1432
<code>\@gls@noexpand@field</code>	26, 89, 90, 478, 510, 728, 760, 761, 982, 1016, 1017, 1137, 1171, 1292, 1328	<code>\@ACRlong</code>	171, 271, 449, 558, 604, 648, 809, 856, 901, 1063, 1110, 1219, 1266, 1376, 1431
<code>\@gls@noidx@no@sanitizesort</code>	28, 479, 729, 983, 1138, 1293	<code>\@ACRlongpl</code>	173, 174, 272, 559, 605, 810, 857, 1064, 1110, 1220, 1266, 1377, 1431
<code>\@gls@noidx@nosanitizesort</code>	27, 208, 479, 579, 729, 831, 983, 1086, 1138, 1242, 1292, 1399	<code>\@ACRshort</code>	167, 168, 271, 448, 555, 604, 647, 807, 856, 901, 1061, 1110, 1217, 1266, 1374, 1431
<code>\@gls@noidx@sanitizesort</code>	27, 479, 728, 983, 1137, 1292	<code>\@ACRshortpl</code>	169, 272, 556, 557, 605, 808, 857, 1062, 1110, 1218, 1266, 1375, 1431
<code>\@gls@nosanitizesort</code>	27, 27, 208, 478, 479, 579, 728, 831, 982, 983, 1086, 1137, 1242, 1292, 1399	<code>\@Acrfull</code>	272, 604, 856, 1111, 1266, 1432
<code>\@gls@old@noidx@nosanitizesort</code>	28, 28, 1293	<code>\@Acrfullpl</code>	273, 604, 856, 1111, 1267, 1432
<code>\@gls@sanitizesort</code>	27, 27, 208, 478, 479, 579, 728, 831, 982, 983, 1086, 1137, 1242, 1292, 1399	<code>\@Acrlong</code>	171, 271, 448, 557, 604, 648, 809, 856, 901, 1063, 1110, 1218, 1219, 1266, 1375, 1376, 1431
<code>\@gls@xdycheckbackslash</code> . .	142, 545, 796, 797, 1050, 1205, 1362	<code>\@Acrlongpl</code>	173, 272, 558, 604, 810, 856, 1064, 1110, 1220, 1266, 1377, 1431
<code>\@gls@xdycheckquote</code>	141, 142, 544, 796, 1049, 1050, 1204, 1205, 1361, 1362	<code>\@Acrshort</code>	167, 447, 555, 647, 807, 900, 1061, 1216, 1373
		<code>\@Acrshortpl</code>	169, 556, 808, 1062, 1217, 1374

\@Alpha	214,	\@GLSuseri@	161, <u>452</u> , 553,
218, 219, 582, 583, 834, 835,		651, 804, 904, 1058, 1214, 1371	
1089–1091, 1245, 1246, 1401, 1403		\@GLSuserii	
\@GLS	147,	. 162, 553, 805, 1058, 1214, 1371	
547, 798, 799, 1052, 1208, 1365		\@GLSuserii@	162, <u>453</u> , 553,
\@GLS@	147, <u>147</u> , 427, 547,	651, 805, 904, 1058, 1214, 1371	
687, 799, 940, 1052, 1208, 1365		\@GLSuseriii	
\@GLSdesc 163, 553, 805, 1059, 1214, 1371	
. 157, 551, 803, 1056, 1212, 1369		\@GLSuseriii@	163, <u>453</u> , 553,
\@GLSdesc@	157, <u>451</u> , 551,	651, 805, 904, 1059, 1214, 1371	
650, 803, 903, 1056, 1212, 1369		\@GLSuseriv	
\@GLSdescplural 164, 554, 805, 1059, 1215, 1372	
. 158, 552, 803, 1057, 1212, 1369		\@GLSuseriv@	164, <u>453</u> , 554,
\@GLSdescplural@	158, <u>451</u> , 552,	651, 805, 904, 1059, 1215, 1372	
650, 803, 903, 1057, 1212, 1369		\@GLSserv	
\@GLSfirst 165, 554, 806, 1060, 1215, 1372	
. 153, 550, 801, 1055, 1210, 1367		\@GLSserv@	165, <u>454</u> , 554,
\@GLSfirst@	153, <u>450</u> , 550,	652, 806, 905, 1060, 1215, 1372	
649, 801, 902, 1055, 1210, 1367		\@GLSservi	
\@GLSfirstplural 166, 554, 806, 1060, 1216, 1373	
. 155, 550, 802, 1056, 1211, 1368		\@GLSservi@	166, <u>454</u> , 554,
\@GLSfirstplural@	155, <u>450</u> , 551,	652, 806, 905, 1060, 1216, 1373	
649, 802, 903, 1056, 1211, 1368		\@GLs	146, 546, 798, 1052, 1207, 1364
\@GLSname		\@GLs@	118, 120, 146, <u>146</u> , 425, 426,
. 156, 551, 802, 1056, 1212, 1369		527, 528, 546, 686, 778, 779,	
\@GLSname@	156, <u>451</u> , 551,	798, 939, 1035, 1036, 1052,	
650, 803, 903, 1056, 1212, 1369		1190, 1191, 1207, 1347, 1348, 1364	
\@GLSpl	149,	\@GLs@link	130, 131, 1041, 1196, 1353
<u>149</u> , 548, 800, 1053, 1209, 1366		\@GLs@acentryname	
\@GLSpl@	149, 428, 548,	<u>176</u> , 274, 560, 605, 811, 857,	
688, 800, 941, 1053, 1209, 1366		1066, 1112, 1221, 1267, 1378, 1433	
\@GLSplural		\@GLs@entry@field	
. 154, 550, 802, 1055, 1211, 1368	 175, <u>175</u> , 176, 177,	
\@GLSplural@	154, <u>450</u> , 550,	515, 559–563, 765, 811–814,	
649, 802, 902, 1055, 1211, 1368		1065, 1066, 1221, 1222, 1378, 1379	
\@GLSsymbol		\@GLs@entryname	176, <u>176</u> ,
. 159, 552, 804, 1057, 1213, 1370		274, 560, 605, 811, 857, 1065,	
\@GLSsymbol@	159, <u>452</u> , 552,	1066, 1112, 1221, 1267, 1378, 1433	
650, 804, 903, 1057, 1213, 1370		\@GLsSetXdyFirstLetterAfterDigits	
\@GLSsymbolplural 188,	
. 160, 552, 804, 1058, 1213, 1370		565, 817, 1071, 1072, 1227, 1384	
\@GLSsymbolplural@	160, <u>452</u> , 552,	\@GLsSetXdyNumberGroupOrder ...	
650, 804, 904, 1058, 1213, 1370	 188, 189, 565, 566,	
\@GLStext	151,	817, 1072, 1227, 1228, 1384, 1385	
152, 549, 801, 1054, 1210, 1367		\@GLsdesc	156,
\@GLStext@	152, <u>449</u> , 549,	157, 551, 803, 1056, 1212, 1369	
649, 801, 902, 1054, 1210, 1367		\@GLsdesc@	157, <u>451</u> , 551,
\@GLSuseri		650, 803, 903, 1056, 1212, 1369	
. 161, 553, 804, 1058, 1214, 1371		\@GLsdescplural	157,
		158, 551, 803, 1057, 1212, 1369	

\@Glsdescplural@ .	158, 451 , 551, 650, 803, 903, 1057, 1212, 1369
\@Glsdisp .	151, 151 , 1054, 1209, 1366
\@Glsentryfield	97, 175 , 177–183, 422, 423, 1021, 1065–1069, 1176, 1221–1225, 1333, 1378–1382
\@Glsfirst	152, 153, 550, 801, 1055, 1210, 1367
\@Glsfirst@	153, 450 , 550, 649, 801, 902, 1055, 1210, 1367
\@Glsfirstplural	154, 155, 550, 802, 1055, 1211, 1368
\@Glsfirstplural@	155, 450 , 550, 802, 1055, 1211, 1368
\@Glsname	155, 156, 551, 802, 1056, 1211, 1368
\@Glsname@	156, 451 , 551, 649, 802, 903, 1056, 1211, 1368
\@Glspl 148, 548, 799, 1053, 1208, 1365	
\@Glspl@ 118, 121, 148, 149 , 426, 527,	529, 548, 686, 687, 778, 779, 799, 940, 1035, 1037, 1053, 1190, 1191, 1208, 1347, 1348, 1365
\@Glsplural	153, 154, 550, 802, 1055, 1211, 1368
\@Glsplural@	154, 450 , 550, 649, 802, 902, 1055, 1211, 1368
\@Glsymbol	158, 159, 552, 803, 1057, 1213, 1370
\@Glsymbol@	159, 452 , 552, 650, 803, 903, 1057, 1213, 1370
\@Glsymbolplural	159, 160, 552, 804, 1057, 1213, 1370
\@Glsymbolplural@ 160, 452 , 552,	650, 804, 904, 1057, 1213, 1370
\@Glstext 152, 549, 801, 1054, 1210, 1367
\@Glstext@	152, 449 , 549, 649, 801, 902, 1054, 1210, 1367
\@Glsuser@i	452–454, 651, 652, 904, 905
\@Glsuseri	160, 161, 553, 804, 1058, 1213, 1370
\@Glsuseri@	161, 452 , 553, 804, 1058, 1213, 1370
\@Glsuserii	161, 162, 553, 805, 1058, 1214, 1371
\@Glsuserii@	162, 453 , 553, 805, 1058, 1214, 1371
\@Glsuseriii	162, 163, 553, 805, 1059, 1214, 1371
\@Glsuseriii@	163, 453 , 553, 805, 1059, 1214, 1371
\@Glsuseriv	163, 164, 554, 805, 1059, 1215, 1372
\@Glsuseriv@	164, 453 , 554, 805, 1059, 1215, 1372
\@Glsuserv	164, 165, 554, 806, 1059, 1215, 1372
\@Glsuserv@	165, 454 , 554, 806, 1059, 1215, 1372
\@Glsuservi	165, 166, 554, 806, 1060, 1215, 1372
\@Glsuservi@	166, 454 , 554, 806, 1060, 1215, 1372
\@Mi	320, 697, 951
\@PGLS	427, 427 , 687, 940
\@PGLS@	427, 427 , 687, 940
\@PGLSpl	427, 427 , 687, 940
\@PGLSpl@	427, 427 , 687, 940
\@Pgls	425, 425 , 686, 939
\@Pgls@	425, 425 , 686, 939
\@Pglspl	426, 426 , 686, 939
\@Pglspl@	426, 426 , 686, 939
\@Roman	215, 218, 219, 582, 583, 834, 835, 1089, 1090, 1245, 1246, 1402, 1403
\@acrfull	272, 272 , 603, 855, 1110, 1266, 1431, 1432
\@acrfullpl 273, 604, 856, 1111, 1267, 1432
\@acrlong 170, 271, 448 , 557, 603,	648, 808, 809, 855, 901, 1062, 1063, 1110, 1218, 1266, 1375, 1431
\@acrlongpl	172, 272, 558, 604, 810, 856, 1064, 1110, 1219, 1266, 1376, 1431
\@acrshort 166, 271, 447 , 555, 603,	604, 647, 806, 855, 856, 900, 1060, 1110, 1216, 1266, 1373, 1431
\@acrshortpl	168, 272, 556, 604, 807, 856, 1061, 1110, 1217, 1266, 1374, 1431
\@addtoacronymlists 23, 476, 726, 980, 1135, 1290
\@addtoacronymlists	23
\@afterheading	308, 404, 410, 674, 691, 927, 945
\@alph	214, 218, 219, 582, 583, 834, 835, 1089–1091, 1245, 1246, 1401, 1403

`\@arabic` 215, 218, 582, 583, 834, 835,
1089, 1090, 1245, 1246, 1401, 1403
`\@auxout` 10, 11, 79, 81, 119, 143, 203,
204, 206, 209, 223, 228, 263,
264, 302, 505, 528, 576–578,
580, 586, 588, 688, 755, 756,
778, 779, 828–830, 832, 838,
840, 941, 1011, 1012, 1035,
1036, 1083, 1085, 1087, 1093,
1095, 1096, 1125, 1165, 1166,
1190, 1206, 1239, 1240, 1243,
1249, 1251, 1322, 1323, 1347,
1363, 1395–1397, 1399, 1406, 1408
`\@backslashchar`
135, 136, 142, 197, 539, 545,
790, 791, 796, 797, 824, 1044,
1045, 1050, 1078, 1199, 1200,
1205, 1234, 1356, 1357, 1362, 1391
`\@bsphack`
. 213, 581, 833, 1088, 1244, 1401
`\@cGls` 120,
120, 528, 779, 1036, 1191, 1348
`\@cGls@` 118,
120, 120, 527, 528, 778, 779,
1035, 1036, 1190, 1191, 1347, 1348
`\@cGlspl` 121, 528, 779, 1036, 1191, 1348
`\@cGlspl@` 118,
121, 121, 527–529, 778, 779,
1035, 1036, 1190, 1191, 1347, 1348
`\@cclv` 320, 697, 698, 951
`\@cgls` 119,
119, 528, 779, 1036, 1190, 1347
`\@cgls@` 117,
119, 119, 527, 528, 777, 779,
1034, 1036, 1189, 1191, 1346, 1348
`\@cglspl` 120, 120, 121,
528, 779, 1036, 1191, 1348
`\@cglspl@` 118,
120, 120, 527, 528, 778, 779,
1035, 1036, 1189, 1191, 1346, 1348
`\@chapter` 50,
489, 739, 995, 1150, 1306, 1307
`\@classoptionslist`
... 46, 488, 738, 994, 1148, 1304
`\@closegls` ... 200, 200, 573, 574,
825, 826, 1080, 1081, 1236, 1393
`\@colht` 320, 698, 951
`\@colroom` 320, 321, 698, 951
`\@currentlabelname`
. 11, 254, 471, 596, 721, 848,
975, 1103, 1130, 1259, 1284, 1424
`\@curroptions`
... 46, 488, 738, 994, 1148, 1304
`\@declaredoptions`
... 46, 488, 738, 994, 1148, 1304
`\@delimN` 602, 854
`\@delimR` 602, 854
`\@disable@onlypremakeg`
. 50, 204, 489, 576, 740, 828,
995, 1083, 1150, 1239, 1307, 1396
`\@disable@premakecs`
50, 50, 490, 740, 995, 1150, 1307
`\@disabled@glssaddxdycounters` ..
..... 63, 64, 496, 746,
747, 1002, 1156, 1157, 1313, 1314
`\@do@addcounter` 62,
63, 495, 496, 746, 1001, 1156, 1313
`\@do@auxoutstuff` 227, 228,
588, 840, 1095, 1096, 1251, 1408
`\@do@glossentry` .. 598, 632, 850, 885
`\@do@glsc@checkesc`
..... 136, 791, 1045, 1200, 1357
`\@do@glsc@getcounterprefix`
..... 220, 582, 583,
834, 835, 1091, 1092, 1247, 1404
`\@do@glsc@islistofacronyms`
... 23, 476, 726, 980, 1135, 1290
`\@do@glscsee`
. 109, 522, 773, 1029, 1184, 1341
`\@do@ifinlist`
... 62, 495, 745, 1001, 1156, 1313
`\@do@newglossaryentry` . 275, 605,
615–628, 661–665, 857, 867–
882, 914–918, 1112, 1268, 1433
`\@do@seeglossary` . 206, 221, 222,
577, 585, 829, 836, 837, 1085,
1092, 1093, 1240, 1248, 1397, 1405
`\@do@subglossentry` 599, 632, 851, 885
`\@do@wrglossary` 132, 214,
538, 581, 668, 788, 833, 921,
1042, 1088, 1197, 1244, 1354, 1401
`\@do@write@glslabels` 38, 39, 485,
735, 989, 990, 1144, 1145, 1300
`\@do@writeaux@info`
. 223, 586, 838, 1093, 1249, 1406
`\@doendpe` .. 227, 404, 410, 1251, 1408
`\@domakeglossaries`
..... 37, 38, 203, 205, 484,
485, 576, 577, 734, 735, 828,

829, 989, 1083, 1084, 1143,	\@firstofthree 129, 145, 148, 150,
1144, 1238, 1240, 1299, 1395, 1397	166, 168, 170, 172, 447–449,
\@ehc 320, 697, 951	536, 546, 547, 549, 555–558,
\@empty 15, 23, 42, 45, 46,	647, 648, 787, 798–800, 806,
50, 62, 64, 67, 70, 71, 105, 110,	808–810, 900–902, 1041, 1051,
111, 133, 134, 145–149, 166–	1053, 1054, 1060, 1061, 1063,
174, 189, 192, 194, 200–202,	1064, 1195, 1207–1209, 1216–
212, 213, 221, 264, 447–449,	1219, 1352, 1364–1366, 1373–1376
473, 475, 476, 486, 488, 489,	\@firstoftwo 32–
495, 496, 498, 500, 501, 520,	34, 94, 95, 112, 129, 148, 149,
523, 538, 546–548, 555–559,	168–170, 172–174, 482, 483,
566, 568, 570, 574, 575, 580,	512, 513, 536, 547, 548, 556–
581, 584, 600, 616, 618–625,	559, 731–733, 763, 764, 787,
627, 629, 647, 648, 668, 670–	799, 800, 807, 808, 810, 811,
672, 722, 724, 726, 736, 738,	986, 987, 1019, 1020, 1031,
739, 745, 746, 749, 751, 770,	1041, 1053, 1061, 1062, 1064,
771, 774, 789, 798–800, 806–	1065, 1141, 1142, 1174, 1175,
811, 818, 820, 822, 825–827,	1186, 1195, 1208, 1209, 1217–
832, 833, 836, 852, 868, 870,	1220, 1296, 1297, 1331, 1332,
872, 874, 876–878, 880, 882,	1343, 1352, 1365, 1366, 1374–1377
900–902, 922, 924, 925, 976,	\@for . . . 34, 46, 50, 52, 62, 64, 71,
978–980, 991–995, 1001, 1002,	94, 95, 135, 189–191, 202, 204,
1004, 1006, 1007, 1026, 1027,	205, 214, 222, 236, 237, 276,
1030, 1042, 1043, 1051–1053,	483, 488, 489, 495, 496, 501,
1060–1065, 1072, 1075, 1076,	512, 513, 539, 566, 567, 576,
1080–1082, 1087, 1088, 1092,	577, 582, 585, 589, 606, 616,
1107, 1131, 1133, 1135, 1146,	618, 620, 621, 623, 626, 627,
1148–1150, 1156, 1157, 1159,	629, 668, 688, 689, 733, 738,
1161, 1162, 1181, 1185, 1197,	740, 746, 751, 763, 764, 790,
1198, 1207–1209, 1216–1220,	818, 819, 828, 829, 834, 837,
1228, 1230, 1232, 1236–1238,	840, 841, 858, 868, 871, 872,
1243, 1244, 1247, 1263, 1286,	874, 876, 879, 880, 882, 922,
1290, 1302–1304, 1307, 1313,	941, 942, 987, 994–996, 1001,
1314, 1316, 1318, 1319, 1338,	1002, 1007, 1019, 1020, 1044,
1342, 1354, 1355, 1364–1366,	1072–1074, 1082–1084, 1089,
1373–1377, 1385, 1387, 1389,	1093, 1096, 1113, 1125, 1142,
1393, 1394, 1400, 1401, 1404, 1428	1148, 1150, 1151, 1156, 1157,
\@end@fixbraces 221, 222,	1161, 1162, 1174, 1175, 1199,
585, 837, 1092, 1093, 1248, 1405	1228, 1229, 1238, 1239, 1244,
\@end@fortrue 34, 75, 95, 483,	1248, 1252, 1269, 1297, 1304,
503, 513, 688, 733, 753, 764,	1307, 1313, 1314, 1318, 1319,
941, 987, 1008, 1020, 1125,	1331, 1332, 1356, 1385, 1386,
1142, 1163, 1175, 1297, 1320, 1332	1395, 1396, 1401, 1405, 1413, 1434
\@esphack	\@glo@desc
. 214, 581, 833, 1088, 1244, 1401	. 107, 521, 772, 1028, 1183, 1340
\@expandtwoargs	\@glo@symbol
. . . 46, 488, 738, 994, 1148, 1304	. 108, 522, 773, 1029, 1183, 1340
\@firstofone 29, 38, 306, 479, 480,	\@glo@access 430, 432, 433, 438, 632,
485, 729, 735, 944, 983, 984,	634, 635, 637, 638, 886–888, 891
989, 1138, 1144, 1293, 1294, 1299	\@glo@addchildren 237,

237, 241, 589, 592, 841, 844, 1096, 1099, 1252, 1255, 1413, 1416	\@glo@do@sortentries
\@glo@assign@sortkey . . 203, 204, 207, 255, 577, 578, 596, 828, 830, 848, 1084, 1085, 1104, 1239, 1241, 1259, 1396, 1398, 1425 236, 237, 589, 840, 841, 1096, 1097, 1252, 1253, 1413, 1414
\@glo@autosee 108, 108, 522, 773, 1029, 1184, 1341	\@glo@entry 186, 187, 565, 816, 1071, 1227, 1384
\@glo@autoseehook 109, 109, 522, 773, 1029, 1184, 1341	\@glo@entryprefix
\@glo@check@mkidxrangear 421, 422, 683, 684, 936, 937
. 134, 219, 539, 584, 668, 790, 835, 921, 1043, 1091, 1198, 1247, 1355, 1403	\@glo@entryprefixfirst
\@glo@check@sortallowed . . . 17, 18–21, 205, 208, 473–475, 577, 579, 723–725, 829, 831, 977– 979, 1084, 1086, 1132–1134, 1239, 1242, 1286–1288, 1396, 1399 421, 422, 683, 684, 936, 937
\@glo@childlist	\@glo@entryprefixfirstplural . .
. 237, 589, 841, 1096, 1252, 1413 421, 422, 683, 684, 936, 937
\@glo@counter 85, 104, 107, 508, 519, 521, 758, 770, 772, 1014, 1026, 1028, 1168, 1180, 1183, 1325, 1337, 1340	\@glo@entryprefixplural
\@glo@counterprefix 209, 216, 219– 221, 264, 267, 580, 582–584, 600, 602, 832, 834–836, 852, 854, 1087, 1090–1092, 1107, 1109, 1243, 1245–1247, 1263, 1264, 1400, 1402–1404, 1428, 1430 421, 422, 683, 684, 936, 937
\@glo@default@sorttype . . 17, 17, 207, 239–241, 473, 578, 591, 592, 723, 830, 842–844, 977, 1085, 1098, 1099, 1132, 1241, 1254, 1255, 1286, 1398, 1415, 1416	\@glo@esclabel 110, 111, 523, 524, 774, 775, 1030, 1031, 1185, 1342
\@glo@defaultcounter	\@glo@etext 529–531, 780–782
. 107, 521, 772, 1028, 1183, 1340	\@glo@first 84, 103, 106, 107, 507, 519, 521, 522, 625, 664, 757, 770, 772, 878, 918, 1013, 1026, 1028, 1168, 1180, 1182, 1183, 1325, 1337, 1339, 1340
\@glo@desc 84, 102, 103, 105, 107, 507, 518– 522, 757, 769, 770, 772, 1013, 1025, 1026, 1028, 1168, 1180, 1181, 1183, 1325, 1337, 1338, 1340	\@glo@firstaccess
\@glo@descaccess 430, 433, 633–635, 886–888
. 430, 433, 434, 633–635, 886–888	\@glo@firstplural 84, 103, 106, 107, 507, 519, 521, 664, 757, 770, 772, 918, 1013, 1026, 1028, 1168, 1180, 1182, 1325, 1337, 1339
\@glo@descplural . . 84, 102, 103, 507, 518, 519, 757, 769, 1013, 1025, 1026, 1168, 1180, 1325, 1337	\@glo@firstpluralaccess
\@glo@descpluralaccess 430, 433, 633–635, 886–888
. 431, 433, 434, 633–635, 886–888	\@glo@grabfirst 247, 247, 593, 845, 1100, 1256, 1420
	\@glo@label . . . 86, 88, 96, 97, 99– 109, 116, 184, 185, 416, 422, 433, 434, 509, 514, 516–522, 526, 563, 564, 634–636, 684, 715, 716, 759, 760, 765, 767– 773, 777, 815, 816, 888, 889, 937, 969, 1014–1016, 1021, 1023–1029, 1034, 1070, 1071, 1169, 1170, 1175–1184, 1189, 1225, 1226, 1326, 1327, 1332– 1341, 1346, 1382, 1383, 1452, 1453

\@glo@longaccess	\@glo@see 85, 104,
431, 433, 434, 633–635, 886–888	108, 109, 508, 519, 522, 758,
\@glo@longpl	770, 773, 1014, 1026, 1029,
. 88, 104, 107, 509, 519, 521,	1168, 1180, 1184, 1325, 1337, 1341
615, 617, 619, 621, 622, 625,	\@glo@seeautonumberlist
626, 661, 759, 770, 772, 867,	13, 13, 86, 472, 508, 721, 758,
868, 870, 871, 873, 875, 878,	975, 1014, 1130, 1169, 1285, 1326
879, 914, 1015, 1026, 1028,	\@glo@short 76, 77, 88, 104,
1170, 1181, 1183, 1327, 1338, 1340	107, 503, 509, 519, 521, 664,
\@glo@longpluralaccess	754, 759, 770, 772, 918, 1009,
431, 433, 434, 633–635, 886–888	1015, 1026, 1028, 1164, 1170,
\@glo@name	1181, 1183, 1321, 1327, 1338, 1340
17, 83, 89, 103, 106, 107, 474,	\@glo@shortaccess
507, 510, 519–521, 723, 757, 431, 433, 434, 633–635,
760, 769, 771, 772, 977, 1013,	661–664, 886–888, 914–916, 918
1016, 1025, 1027, 1028, 1132,	\@glo@shortpl
1168, 1171, 1180, 1182, 1183,	88,
1287, 1324, 1328, 1337, 1339, 1340	104, 107, 509, 519, 521, 615,
\@glo@no@assign@sortkey .. 203,	617, 619, 621, 622, 625, 626,
204, 255, 577, 596, 828, 848,	661, 664, 759, 770, 772, 867–
1084, 1104, 1239, 1259, 1396, 1425	869, 871, 873, 875, 878, 879,
\@glo@nonumberlist	914, 918, 1015, 1026, 1028,
87,	1170, 1181, 1183, 1327, 1338, 1340
508, 509, 759, 1015, 1169, 1326	\@glo@shortpluralaccess
\@glo@numfmt	431, 433, 434, 633–635, 886–888
220, 584,	\@glo@sort 17, 20, 27, 28, 84,
668, 836, 921, 1091, 1247, 1404	104, 106, 110, 111, 474, 475,
\@glo@parent	479, 507, 519, 521, 523, 524,
. 86, 104–106, 110, 111, 237,	723, 725, 728, 729, 757, 770,
238, 475, 508, 519, 520, 523,	771, 774, 775, 977–979, 983,
524, 589, 724, 758, 770, 771,	1013, 1026, 1027, 1030, 1031,
774, 775, 841, 978, 1014, 1026,	1132, 1134, 1137, 1138, 1168,
1027, 1030, 1031, 1097, 1133,	1180, 1182, 1185, 1287, 1288,
1169, 1180–1182, 1185, 1252,	1292, 1293, 1325, 1337, 1339, 1342
1253, 1326, 1337–1339, 1342, 1414	\@glo@sortedinsert
\@glo@plural 84, 103, 106, 238, 238, 589, 590,
507, 519–521, 663, 757, 770–	841, 842, 1097, 1252, 1253, 1414
772, 916, 1013, 1026–1028,	\@glo@sortentries
1168, 1180, 1182, 1325, 1337, 1339 235, 236, 236, 588, 591,
\@glo@pluralaccess	592, 840, 842, 843, 1096, 1098,
.... 430, 433, 633–635, 886–888	1099, 1251, 1254, 1255, 1412, 1413
\@glo@prefix	\@glo@sorthandler@case
.. 13, 86, 104, 110, 111, 134,	236, 239, 590, 591, 842, 843,
135, 219, 220, 472, 508, 519,	1097, 1099, 1253, 1254, 1412, 1415
523, 524, 539, 584, 668, 721,	\@glo@sorthandler@letter
758, 770, 774, 775, 790, 836,	235, 238, 590, 591, 842, 843,
921, 976, 1014, 1026, 1030,	1097, 1098, 1253, 1254, 1412, 1414
1031, 1043, 1044, 1091, 1130,	\@glo@sorthandler@nocase
1169, 1180, 1185, 1198, 1247,	236, 239, 590, 592, 842, 843,
1285, 1326, 1337, 1342, 1355, 1404	1098, 1099, 1253, 1255, 1412, 1415
\@glo@range ... 219, 220, 584, 668,	
836, 921, 1091, 1247, 1403, 1404	

<code>\@glo@sorthandler@word</code>	770, 773, 1013, 1026, 1029, 1168, 1180, 1183, 1325, 1337, 1340
. 235, 238, 590, 591, 842, 1097, 1098, 1253, 1254, 1412, 1414	
<code>\@glo@sortinghandler</code>	<code>\@glo@symbolpluralaccess</code>
236, 238, 588, 590, 840, 842, 1096, 1097, 1251, 1253, 1413, 1414 430, 433, 633–635, 886–888
<code>\@glo@sortinglist</code> 236, 238, 241, 588–590, 592, 840, 841, 844, 1096, 1097, 1099, 1251–1253, 1255, 1413, 1414, 1416	<code>\@glo@text</code> 84, 103, 106, 108, 146–150, 175–177, 507, 519, 521, 522, 546–549, 559, 560, 621, 663, 684, 757, 770, 771, 773, 798–800, 811, 812, 873, 916, 937, 1013, 1026, 1027, 1029, 1052–1054, 1065, 1066, 1168, 1180, 1182, 1183, 1207– 1209, 1221, 1222, 1325, 1337, 1339, 1340, 1364–1366, 1378, 1379
<code>\@glo@sortmacro@case</code>	<code>\@glo@textaccess</code> 430, 433, 632, 634, 635, 661–663, 886–888, 914–917
. 240, 591, 843, 1099, 1254, 1416	<code>\@glo@thislabel</code> 109, 523, 773, 774, 1030, 1184, 1341
<code>\@glo@sortmacro@def</code>	<code>\@glo@thislettergrp</code> 235, 245, 247– 249, 593, 594, 845, 846, 1100, 1101, 1256, 1257, 1412, 1419–1421
. 241, 592, 844, 1099, 1255, 1416	<code>\@glo@thisvalue</code> 77, 78, 504, 754, 1010, 1164, 1165, 1321, 1322
<code>\@glo@sortmacro@def@do</code>	<code>\@glo@tmp</code> 79, 96, 97, 221, 514, 584, 765, 836, 1011, 1021, 1092, 1165, 1175, 1176, 1247, 1322, 1332, 1333, 1404
. 241, 592, 844, 1099, 1255, 1416	<code>\@glo@type</code> 11, 85, 103– 105, 107, 108, 185, 186, 204, 211, 212, 225–228, 236, 238, 241–243, 246, 247, 253, 254, 263, 274, 275, 471, 475, 507, 519–522, 564, 565, 576, 580, 581, 587–589, 592, 593, 595, 596, 605, 616, 618, 619, 621, 623, 625, 627, 629, 688, 689, 721, 724, 757, 769–773, 816, 828, 832, 833, 838–841, 844, 845, 847, 848, 857, 868, 870, 872, 874, 876, 878, 880, 882, 941, 942, 975, 978, 979, 1013, 1026–1029, 1071, 1083, 1087, 1088, 1094–1097, 1099, 1100, 1102, 1103, 1112, 1130, 1133, 1168, 1180, 1181, 1183, 1184, 1226, 1227, 1239, 1243, 1250– 1252, 1255, 1256, 1258, 1259, 1267, 1284, 1325, 1337, 1338, 1340, 1341, 1383, 1384, 1396, 1400, 1407, 1408, 1413, 1414,
<code>\@glo@sortmacro@letter</code>	
. 240, 591, 843, 1098, 1254, 1415	
<code>\@glo@sortmacro@lettercase</code> 240	
<code>\@glo@sortmacro@nocase</code>	
. 241, 591, 843, 1099, 1255, 1416	
<code>\@glo@sortmacro@standard</code>	
. 240, 591, 843, 1098, 1254, 1415	
<code>\@glo@sortmacro@use</code>	
. 242, 592, 844, 1100, 1255, 1417	
<code>\@glo@sortmacro@word</code>	
. 239, 590, 842, 1098, 1254, 1415	
<code>\@glo@sortmacro@wordcase</code> 239	
<code>\@glo@sorttype</code> 207, 242, 246, 248, 255, 578, 592, 593, 596, 830, 844, 845, 848, 1085, 1100, 1104, 1241, 1255, 1256, 1259, 1398, 1417, 1420, 1421, 1425	
<code>\@glo@storeentry</code>	
17, 18, 20, 110, 473–475, 523, 723, 724, 774, 977–979, 1030, 1132, 1133, 1185, 1287, 1288, 1342	
<code>\@glo@sufffix</code> 134, 135, 220, 539, 584, 668, 790, 836, 921, 1043, 1044, 1091, 1198, 1247, 1355, 1404	
<code>\@glo@symbol</code> 76, 85, 103, 108, 503, 507, 519, 522, 620, 624, 753, 757, 770, 773, 873, 877, 1009, 1013, 1026, 1029, 1164, 1168, 1180, 1183, 1321, 1325, 1337, 1340	
<code>\@glo@symbolaccess</code> 430, 433, 633–635, 664, 886–888, 917	
<code>\@glo@symbolplural</code> 85, 103, 108, 507, 519, 522, 757,	

1416, 1417, 1420, 1423, 1424, 1433
\@glo@types 39, 71,
79, 81, 115, 116, 186, 187, 202,
204, 205, 298, 415, 485, 501,
505, 525, 526, 565, 576, 577,
631, 715, 735, 751, 755, 776,
777, 816, 828, 829, 884, 969,
990, 1007, 1010, 1011, 1033,
1034, 1071, 1082–1084, 1124,
1144, 1161, 1162, 1165, 1166,
1188, 1227, 1238–1240, 1280,
1300, 1318, 1319, 1322, 1323,
1345, 1384, 1395, 1396, 1445, 1452
\@glo@useri 87,
104, 107, 509, 519, 521, 759,
770, 772, 1015, 1026, 1028,
1170, 1180, 1183, 1327, 1337, 1340
\@glo@useriaccess
431, 433, 434, 633–635, 886–888
\@glo@userii 87,
104, 107, 509, 519, 521, 759,
770, 772, 1015, 1026, 1028,
1170, 1181, 1183, 1327, 1338, 1340
\@glo@useriiaaccess 431,
433, 434, 633–635, 886, 887, 889
\@glo@useriii 87,
104, 107, 509, 519, 521, 759,
770, 772, 1015, 1026, 1028,
1170, 1181, 1183, 1327, 1338, 1340
\@glo@useriiiaaccess 431,
433, 434, 633–635, 886, 887, 889
\@glo@useriv 88,
104, 107, 509, 519, 521, 759,
770, 772, 1015, 1026, 1028,
1170, 1181, 1183, 1327, 1338, 1340
\@glo@userivaccess
431, 433, 434, 633–635, 887–889
\@glo@userv 88,
104, 107, 509, 519, 521, 759,
770, 772, 1015, 1026, 1028,
1170, 1181, 1183, 1327, 1338, 1340
\@glo@uservaccess
432–434, 633, 634, 636, 887–889
\@glo@uservi 88,
104, 107, 509, 519, 521, 759,
770, 772, 1015, 1026, 1028,
1170, 1181, 1183, 1327, 1338, 1340
\@glo@userviaccess
432–434, 633, 634, 636, 887–889
\@glodesc
. 107, 522, 772, 1029, 1183, 1340
\@glolist@
. 105, 520, 771, 1027, 1181, 1338
\@gloname
. 107, 521, 772, 1028, 1183, 1340
\@glossary@default@style
. 12, 12, 14, 225,
264, 293, 471, 472, 587, 601,
629, 721, 722, 839, 853, 882,
975, 976, 1094, 1107, 1122,
1123, 1130, 1131, 1250, 1263,
1278, 1284, 1285, 1407, 1428, 1444
\@glossary@write@seclabel
. 10, 11, 253, 254
\@glossaryentryfield
109, 110, 111, 523, 524, 774,
1030, 1031, 1184, 1185, 1341, 1342
\@glossarylabeltype 10, 11, 11
\@glossarysection
. 58, 60, 493, 494, 743,
744, 999, 1153, 1154, 1310, 1311
\@glossarystyle 225, 226,
253, 587, 595, 839, 847, 1094,
1095, 1103, 1250, 1258, 1407, 1424
\@glossarysubentryfield
. 109, 110, 111, 523, 524,
774, 775, 1030, 1031, 1185, 1342
\@gls 145,
145, 546, 798, 1051, 1207, 1364
\@gls@ 117, 119,
145, 145, 424–426, 527, 528,
546, 685, 686, 778, 779, 798,
938, 939, 1035, 1036, 1051,
1189, 1191, 1207, 1346, 1348, 1364
\@gls@@automake@immediate
. 202, 203, 576,
827, 828, 1082, 1083, 1238, 1395
\@gls@@link 130, 130, 131,
536, 537, 787, 1041, 1196, 1353
\@gls@Hcounter 133, 134, 538, 789,
1042, 1043, 1197, 1198, 1354, 1355
\@gls@ReturnAfterFi 268,
602, 603, 854, 1109, 1265, 1430
\@gls@access@display . . 438, 637, 891
\@gls@actualchar 111, 138, 140, 193,
524, 541, 542, 544, 569, 671,
774, 775, 792, 793, 795, 820,
821, 924, 1030, 1031, 1046,

1048, 1049, 1075, 1185, 1201,
 1203, 1231, 1342, 1358, 1360, 1388
 \@gls@addpredefinedattributes .
 64, 189, 199, 497,
 566, 573, 747, 818, 825, 1002,
 1003, 1072, 1080, 1157, 1158,
 1228, 1235, 1314, 1315, 1385, 1392
 \@gls@adjustmode 186,
 186, 564, 816, 1071, 1226, 1383
 \@gls@after
 23, 476, 477, 726, 981, 1135, 1290
 \@gls@assign@symbol@field 27
 \@gls@assign@symbolplural@field 27
 \@gls@automake ... 200, 205, 573,
 574, 577, 825, 826, 829, 1080,
 1081, 1084, 1236, 1239, 1393, 1396
 \@gls@automake@immediate
 . 201, 202, 574–576, 826–828,
 1081, 1082, 1237, 1238, 1394, 1395
 \@gls@automake@types 202, 202, 205,
 1082, 1084, 1238, 1239, 1395, 1396
 \@gls@before
 ... 23, 476, 726, 981, 1135, 1290
 \@gls@between
 302, 303, 688, 689, 942, 1125, 1126
 \@gls@body 177, 560, 811,
 812, 1066, 1221, 1222, 1378, 1379
 \@gls@checkactual
 136, 140, 197, 540,
 544, 571, 791, 795, 823, 1045,
 1048, 1049, 1078, 1199, 1203,
 1204, 1234, 1356, 1360, 1361, 1391
 \@gls@checkboxar 136, 139, 197, 540,
 543, 572, 791, 794, 823, 1045,
 1047, 1048, 1078, 1199, 1202,
 1203, 1234, 1356, 1359, 1360, 1391
 \@gls@checkedmkidx 135–
 142, 196–198, 539–545, 570–
 572, 790–797, 822–824, 1044–
 1050, 1077–1079, 1199–1205,
 1233–1235, 1356–1362, 1390–1392
 \@gls@checkesc
 136, 141, 197, 791, 824,
 1045, 1078, 1200, 1234, 1357, 1391
 \@gls@checkescactual 136,
 137, 197, 540–542, 571, 791–
 793, 823, 1044, 1046, 1078,
 1199, 1201, 1233, 1356, 1358, 1390
 \@gls@checkescbar 136, 138, 197,
 540, 542, 572, 791, 793, 823,
 1045–1047, 1078, 1199, 1201,
 1202, 1234, 1356, 1358, 1359, 1391
 \@gls@checkesclevel 136,
 138, 197, 542, 543, 791, 793,
 794, 824, 1045, 1047, 1078,
 1200, 1202, 1234, 1357, 1359, 1391
 \@gls@checkescquote 136,
 137, 197, 198, 540, 541, 571–
 573, 791, 792, 823, 824, 1044–
 1046, 1078, 1079, 1199–1201,
 1233–1235, 1356–1358, 1390–1392
 \@gls@checklevel 136, 140,
 197, 540, 543, 572, 791, 794,
 795, 823, 1045, 1048, 1078,
 1200, 1203, 1234, 1357, 1360, 1391
 \@gls@checkmkidxchars
 110, 134, 135, 196, 206,
 219, 221, 523, 539, 540, 571,
 577, 583, 585, 668, 774, 790,
 791, 823, 829, 835, 836, 921,
 1030, 1043, 1044, 1078, 1084,
 1091, 1092, 1185, 1198, 1199,
 1233, 1240, 1246, 1248, 1342,
 1355, 1356, 1390, 1397, 1403, 1405
 \@gls@checkquote
 136, 136, 196, 540, 541, 570,
 571, 791, 792, 822, 823, 1044,
 1045, 1077, 1078, 1199, 1200,
 1232, 1233, 1356, 1357, 1389, 1390
 \@gls@classI 190, 566, 567,
 818, 1073, 1228, 1229, 1385, 1386
 \@gls@classII
 . 190, 567, 818, 1073, 1229, 1386
 \@gls@codepage
 . 69, 228, 500, 588, 750, 840,
 1006, 1096, 1161, 1251, 1318, 1408
 \@gls@counter 128,
 132–134, 185, 186, 209, 220,
 221, 536–539, 564, 580, 584,
 668, 787–789, 816, 832, 836,
 921, 1040, 1042, 1043, 1071,
 1087, 1091, 1092, 1195, 1197,
 1198, 1226, 1243, 1247, 1248,
 1352, 1354, 1355, 1383, 1400, 1404
 \@gls@counterwithin
 15, 16, 16, 473, 722,
 723, 976, 977, 1131, 1132, 1286
 \@gls@ctr 62, 495, 746, 1001, 1156, 1313
 \@gls@currentglossaryxrlabel ..
 11, 11, 253, 254, 263

<code>\@gls@currentlettergroup</code>	1132–1134, 1183, 1287, 1288, 1340
. 247–249, 592, 594, 844, 846,	
1100, 1101, 1255, 1257, 1420, 1421	
<code>\@gls@debugfalse</code>	17–20, 81, 474, 475, 506,
. . . . 7, 469, 719, 973, 1127, 1282	723–725, 756, 977–979, 1012,
<code>\@gls@debugtrue</code> 7, 469, 718,	1132–1134, 1166, 1287, 1288, 1323
719, 972, 973, 1127, 1128, 1282	
<code>\@gls@declareoption</code>	<code>\@gls@deprecated@acr@warning</code> . .
. . . 12, 13, 14, 21, 22, 25, 26, 291, 870, 872, 874,
33, 36–39, 41, 42, 44, 45, 471,	876, 878, 880, 1122, 1278, 1443
472, 475–478, 482, 484, 485,	<code>\@gls@do@acronymsdef</code>
487, 488, 721, 722, 725–728, 22, 22, 49, 82, 475,
732, 734, 735, 737, 738, 975,	476, 489, 506, 725, 726, 739,
976, 979–982, 986, 988–991,	756, 980, 994, 995, 1012, 1134,
993, 1130, 1131, 1134–1137,	1135, 1149, 1167, 1289, 1306, 1324
1141, 1143–1148, 1284, 1285,	<code>\@gls@do@automake@err</code> 40, 41, 42,
1289, 1291, 1296, 1298–1301, 1303	990, 991, 1145, 1146, 1300, 1301
<code>\@gls@default</code> 122, 529,	<code>\@gls@do@glunset</code> 129, 129,
780, 1037, 1191, 1192, 1348, 1349	146–150, 1040, 1052–1054,
<code>\@gls@default@value</code>	1195, 1207–1209, 1352, 1364–1366
. 76–78, 84, 89, 90, 103, 104,	<code>\@gls@do@indexdef</code>
106, 107, 291, 421, 422, 503,	. . . 45, 49, 82, 488, 489, 506,
504, 507, 510, 511, 519–522,	738, 739, 756, 993–995, 1012,
624, 683, 684, 753, 754, 757,	1148, 1149, 1167, 1303, 1306, 1324
760, 761, 769–772, 877, 937,	<code>\@gls@do@numbersdef</code>
1009, 1010, 1013, 1016, 1017,	. . . 44, 49, 82, 487–489, 506,
1026–1029, 1122, 1164, 1165,	738, 739, 756, 993–995, 1012,
1168, 1171, 1172, 1180, 1182,	1148, 1149, 1167, 1303, 1306, 1324
1183, 1278, 1321, 1322, 1325,	<code>\@gls@do@symbolsdef</code>
1328, 1329, 1337, 1339, 1340, 1443	. . 44, 82, 487, 506, 737, 756,
<code>\@gls@defaulttype</code>	993, 1012, 1147, 1167, 1303, 1324
. 79, 79, 1011, 1165, 1322	<code>\@gls@do@symbolssdef</code>
<code>\@gls@deffile</code>	49, 489, 739, 994, 995, 1149, 1306
. . 92–94, 512, 513, 762, 763,	<code>\@gls@doautomake</code> 41, 41, 42,
1018, 1019, 1173, 1174, 1330, 1331	202, 203, 205, 486, 576, 577,
<code>\@gls@define@glossaryentrycounter</code>	736, 828, 829, 991, 1083, 1084,
. 15, 16, 49, 254,	1146, 1238, 1239, 1301, 1395, 1396
256, 473, 489, 596, 597, 722,	<code>\@gls@docheckquotedef</code>
723, 739, 848, 849, 976, 977,	. 196–198, 570–573, 822–825,
994, 1103, 1104, 1131, 1149,	1077–1079, 1232–1235, 1389–1392
1259, 1260, 1286, 1306, 1424, 1425	<code>\@gls@docloadedfalse</code>
<code>\@gls@define@glossarysubentrycounter</code> 6, 468, 718, 972, 1127, 1281
. 16,	<code>\@gls@docloadedtrue</code>
49, 254, 256, 473, 489, 596, 6, 468, 718, 972, 1127, 1281
597, 723, 739, 848, 849, 977,	<code>\@gls@dodeflistparser</code> 204,
995, 1103, 1104, 1132, 1149,	577, 828, 1083, 1084, 1239, 1396
1259, 1260, 1286, 1306, 1424, 1425	<code>\@gls@doentrycounterdef</code>
<code>\@gls@defsort</code>	49, 489, 739, 994, 995, 1149, 1306
. . 17–20, 108, 474, 475, 522,	<code>\@gls@doentrydef</code> . 535, 536, 786, 787
723–725, 773, 977–979, 1029,	<code>\@gls@dolast</code> 222,
	585, 837, 1093, 1248, 1249, 1405

<code>\@gls@donext</code>	222,	1022, 1054–1060, 1176, 1177,
	585, 837, 1093, 1248, 1249, 1405	1210–1216, 1333, 1334, 1367–1373
<code>\@gls@donext@def</code>	184, 185,	<code>\@gls@fieldaccess@display</code> .
	564, 815, 816, 1070, 1226, 1383	438,
<code>\@gls@dodosubentrycounterdef</code>		438, 439–441, 638–640, 891–893
	. . . 49, 489, 739, 995, 1149, 1306	<code>\@gls@firsttok</code>
<code>\@gls@dothiswrite</code>	574, 575,	. 247, 593, 845, 1100, 1256, 1420
	826, 827, 1080–1082, 1236–1238	<code>\@gls@fixbraces</code>
<code>\@gls@elem</code>	688, 941, 1125	109, 221, 522, 585, 773, 837,
<code>\@gls@enablesavenonumberlist</code> . .		1029, 1092, 1184, 1248, 1341, 1405
	. . 87, 93, 508, 512, 759, 763,	<code>\@gls@forbidtexext</code> . .
	1015, 1019, 1169, 1173, 1326, 1330	79, 81, 505,
<code>\@gls@encapchar</code>	138,	755, 756, 1011, 1166, 1322, 1323
	139, 193, 220, 221, 542, 543,	<code>\@gls@fulllist</code> 39, 735, 990, 1144, 1300
	569, 584, 585, 668, 671, 793,	<code>\@gls@get@counterprefix</code>
	794, 820, 821, 836, 837, 921,	. 221, 584, 836, 1092, 1247, 1404
	924, 1047, 1048, 1075, 1091,	<code>\@gls@getbody</code>
	1092, 1201–1203, 1231, 1247,	. 177, 560, 811, 1066, 1221, 1378
	1248, 1358–1360, 1388, 1404, 1405	<code>\@gls@getcounter</code>
<code>\@gls@entry@count</code> 82, 506, 756, 1012, 1167, 1324
 118, 119, 119, 527, 528,	<code>\@gls@getcounterprefix</code>
	778, 779, 1035, 1036, 1190, 1347 220, 220, 582–
<code>\@gls@entry@field</code> 96, 97, 117, 143,		584, 834–836, 1092, 1247, 1404
	174, 175–184, 434–436, 514,	<code>\@gls@getgrouptitle</code>
	515, 526, 559–563, 636, 637,	206, 262, 262, 303, 403, 577,
	765, 777, 811–815, 889, 890,	600, 689, 829, 852, 942, 1085,
	1021, 1034, 1065–1069, 1175,	1106, 1126, 1240, 1262, 1397, 1427
	1176, 1189, 1206, 1220–1225,	<code>\@gls@glossary</code> 212, 212, 213, 581,
	1332, 1333, 1346, 1363, 1377–1382	833, 1088, 1243, 1244, 1400, 1401
<code>\@gls@escbsdq</code>		<code>\@gls@glslink@hyper@update@hook</code>
	135, 136, 194, 539, 540, 570, 128, 128, 1040, 1195, 1352
	672, 790, 791, 821, 822, 925,	<code>\@gls@gobbleopt</code>
	1044, 1076, 1199, 1232, 1356, 1389	. . 80, 505, 755, 1011, 1166, 1323
<code>\@gls@expand@fields</code>		<code>\@gls@grptitle</code> 262, 300,
 90, 90, 91, 511,	303, 600, 688, 689, 852, 941,
	761, 1017, 1171, 1172, 1328, 1329	942, 1106, 1125, 1126, 1262, 1427
<code>\@gls@expandonce</code>		<code>\@gls@hyp@opt</code> 98,
	. . 91, 511, 761, 1017, 1172, 1329	119–121, 129, 130, 145–173,
<code>\@gls@extramakeindexopts</code>		272, 273, 424–427, 515, 516,
	198, 203, 573, 576, 825, 828,	528, 536, 546–559, 603, 604,
	1079, 1083, 1235, 1239, 1392, 1396	685–687, 766, 779, 787, 798–
<code>\@gls@fetch@lettergrp</code>	235,	810, 855, 856, 938–940, 1022,
	236, 245, 248, 1412, 1419, 1421	1036, 1041, 1051–1064, 1110,
<code>\@gls@fetchfield</code>		1111, 1176, 1177, 1190, 1191,
	. . 78, 95, 504, 513, 754, 764,	1195, 1196, 1207–1220, 1266,
	1010, 1020, 1165, 1175, 1322, 1332	1267, 1333, 1334, 1347, 1348,
<code>\@gls@field@link</code>		1352, 1353, 1364–1377, 1431, 1432
	. 98, 151, 151, 152–166, 449–	<code>\@gls@hyp@opt@cs</code> . 129, 130, 536,
	454, 515, 516, 549–555, 649–	787, 1041, 1195, 1196, 1352, 1353
	652, 766, 801–806, 902–905,	<code>\@gls@hypergroup</code>
	 302, 302, 688, 941, 1125

<code>\@gls@ifmeasuring</code>	111 ,	798–800 , 1041 , 1051–1054 ,
	112 , 1031 , 1185 , 1186 , 1342 , 1343	1196 , 1207–1209 , 1353 , 1364–1366
<code>\@gls@ifinlist</code>	62 ,	<code>\@gls@link@label</code> 132 , 537 , 617 , 622 ,
	62 , 495 , 745 , 746 , 1001 , 1156 , 1313	788 , 869 , 875 , 1042 , 1197 , 1354
<code>\@gls@ifnotmeasuring</code>	524 , 775	<code>\@gls@link@nocheckfirsthyper</code> . .
<code>\@gls@igtype</code> 151 , 151 ,
	. . 83 , 507 , 757 , 1013 , 1167 , 1324	166–174 , 549 , 555–559 , 801 ,
<code>\@gls@increment@currcount</code>		806–810 , 1054 , 1060–1065 ,
	117 , 118 , 526 , 527 , 777 , 778 ,	1210 , 1216–1220 , 1367 , 1373–1377
	1034 , 1035 , 1189 , 1190 , 1346 , 1347	<code>\@gls@link@opts</code> 132 , 537 , 617 , 622 ,
<code>\@gls@indexdef</code>		788 , 869 , 875 , 1042 , 1197 , 1354
	45 , 45 , 488 , 738 , 993 , 1148 , 1303	<code>\@gls@list</code> . . 688 , 689 , 941 , 942 , 1125
<code>\@gls@initnonumberlist</code>		<code>\@gls@listsuffix</code>
 87 , 87 , 104 ,	. . 62 , 495 , 745 , 1001 , 1156 , 1313
	508 , 519 , 758 , 759 , 770 , 1014 ,	<code>\@gls@loadlist</code> 14 , 14 ,
	1015 , 1026 , 1169 , 1180 , 1326 , 1337	292 , 472 , 629 , 722 , 882 , 976 ,
<code>\@gls@islistofacronyms</code>		1122 , 1130 , 1131 , 1278 , 1285 , 1443
	. . . 23 , 476 , 726 , 980 , 1135 , 1290	<code>\@gls@loadlong</code> . . 13 , 13 , 14 , 292 ,
<code>\@gls@keylist</code>	465 , 661 , 914	472 , 629 , 721 , 722 , 882 , 976 ,
<code>\@gls@keymap</code>	87 , 94 ,	1122 , 1130 , 1131 , 1278 , 1285 , 1443
	94 , 95–97 , 421 , 432 , 509 , 512–	<code>\@gls@loadshellesc</code> 47–49 , 1304–1306
	514 , 634 , 683 , 759 , 763–765 ,	<code>\@gls@loadsuper</code> 14 , 14 ,
	887 , 936 , 1015 , 1019–1021 ,	292 , 472 , 629 , 722 , 882 , 976 ,
	1170 , 1174–1176 , 1327 , 1331–1333	1122 , 1130 , 1131 , 1278 , 1285 , 1443
<code>\@gls@label</code>	206 , 209 , 216 ,	<code>\@gls@loadtree</code> 14 ,
	219 , 220 , 577 , 578 , 580 , 583 ,	14 , 292 , 472 , 629 , 722 , 882 ,
	584 , 829 , 832 , 834–836 , 1085 ,	976 , 1122 , 1131 , 1278 , 1285 , 1443
	1087 , 1090 , 1091 , 1240 , 1243 ,	<code>\@gls@local@increment@currcount</code>
	1246 , 1247 , 1397 , 1400 , 1402–1404	117 , 118 , 526 , 527 , 777 , 778 ,
<code>\@gls@labelsfile</code>	39 ,	1034 , 1035 , 1189 , 1190 , 1346 , 1347
	485 , 735 , 990 , 1144 , 1145 , 1300	<code>\@gls@loclist</code> . 207 , 208 , 244 , 246 ,
<code>\@gls@langmod</code>	37 , 200 , 201 ,	248 , 249 , 578 , 579 , 593 , 594 ,
	228 , 574 , 575 , 826 , 827 , 988 ,	830 , 831 , 845 , 846 , 1085 , 1086 ,
	1080 , 1081 , 1095 , 1143 , 1236 ,	1101 , 1241 , 1242 , 1256 , 1257 ,
	1237 , 1251 , 1298 , 1393 , 1394 , 1408	1398 , 1399 , 1417–1419 , 1421 , 1422
<code>\@gls@levelchar</code> 111 , 139 , 140 , 193 ,		<code>\@gls@makeindexapp</code>
524 , 542 , 543 , 569 , 671 , 775 ,		. . . 16 , 36 , 36 , 195 , 200 , 202–204
793 , 794 , 820 , 821 , 924 , 1031 ,		<code>\@gls@map</code> 94 , 95 , 512 , 513 , 763 , 764 ,
1047 , 1048 , 1075 , 1185 , 1202 ,		1019 , 1020 , 1174 , 1175 , 1331 , 1332
1203 , 1231 , 1342 , 1359 , 1360 , 1388		<code>\@gls@missinglang@warn</code>
<code>\@gls@link</code>	130 ,	25 , 25 , 53 , 478 , 491 , 727 , 741 ,
	132 , 146–151 , 167–174 , 447–	982 , 997 , 1136 , 1151 , 1291 , 1308
	449 , 537 , 546–549 , 555–559 ,	<code>\@gls@missingnumberlist</code>
	647 , 648 , 787 , 788 , 798–801 ,	. 89 , 107 , 510 , 521 , 760 , 772 ,
	806–811 , 900–902 , 1041 , 1042 ,	1016 , 1028 , 1170 , 1183 , 1327 , 1340
	1052–1054 , 1060–1065 , 1196 ,	<code>\@gls@noaccess</code>
	1197 , 1207–1210 , 1216–1220 ,	432 , 438 , 634 , 637 , 638 , 887 , 891
	1353 , 1354 , 1364–1367 , 1373–1377	<code>\@gls@noexpand@field</code> 89
<code>\@gls@link@checkfirsthyper</code> 131 ,		<code>\@gls@noexpand@fields</code>
145–150 , 537 , 546–549 , 788 ,		. . 89 , 91 , 510 , 511 , 760 , 761 ,

1016, 1017, 1171, 1172, 1328, 1329
 \@gls@nohyperlist 24,
 24, 83, 131, 477, 506, 507, 537,
 727, 757, 788, 981, 1013, 1042,
 1136, 1167, 1196, 1290, 1324, 1353
 \@gls@noidx@addtorefs . 206, 209, 210
 \@gls@noidx@do 247, 248, 593, 844,
 845, 1100, 1101, 1256, 1420, 1421
 \@gls@noidx@getgrouptitle
 206, 262, 577, 600, 829, 852,
 1085, 1106, 1240, 1262, 1397, 1427
 \@gls@noidx@sanitizesort
 . 27, 208, 479, 579, 728, 831,
 983, 1086, 1137, 1242, 1292, 1399
 \@gls@noidx@setsanitizesort
 30, 31, 31, 208,
 481, 579, 730, 731, 831, 985,
 1086, 1139, 1140, 1242, 1295, 1399
 \@gls@noidx@loclist@finalsep
 207, 208,
 579, 830, 831, 1086, 1241, 1398
 \@gls@noidx@loclist@prev
 208, 249, 250, 579,
 594, 595, 831, 846, 847, 1086,
 1102, 1241, 1257, 1258, 1398, 1422
 \@gls@noidx@loclist@sep
 207, 249, 250, 578, 579,
 594, 595, 830, 846, 847, 1086,
 1102, 1241, 1257, 1258, 1398, 1422
 \@gls@noist@toolate 193,
 195, 199, 1075, 1076, 1080,
 1231, 1232, 1235, 1388, 1389, 1392
 \@gls@noref@warn . 206, 209, 243,
 247, 577, 580, 593, 829, 832,
 845, 1084, 1087, 1100, 1240,
 1242, 1256, 1397, 1399, 1417, 1420
 \@gls@notranslatorhook 32
 \@gls@numberlink
 . 267, 602, 854, 1108, 1264, 1429
 \@gls@numbersdef 44,
 45, 487, 488, 738, 993, 1148, 1303
 \@gls@numlist@lastsep . 184, 185,
 564, 815, 816, 1070, 1226, 1383
 \@gls@numlist@nextsep . 184, 185,
 564, 815, 816, 1070, 1226, 1383
 \@gls@numlist@sep 184, 185,
 564, 815, 816, 1070, 1226, 1383
 \@gls@old@chapter 50,
 489, 739, 995, 1150, 1306, 1307
 \@gls@old@print@noidx@glossary
 246, 1420
 \@gls@oldnewglossaryentryposthook
 433, 634, 888
 \@gls@oldnewglossaryentryprehook
 432, 634, 887
 \@gls@onlypremakeg 50,
 50, 489, 739, 740, 995, 1150, 1307
 \@gls@order
 . 200–202, 574, 575, 825–827,
 1080–1082, 1236–1238, 1393–1395
 \@gls@org@LT@output . . . 320, 697, 951
 \@gls@org@glsnoidx@displayloc
 . 208, 579, 831, 1086, 1242, 1399
 \@gls@org@glsseeformat
 . 208, 579, 831, 1086, 1242, 1399
 \@gls@override@glossary
 43, 44, 486, 487,
 737, 992, 993, 1147, 1302, 1303
 \@gls@patchtabularx
 . 113, 524, 775, 1032, 1187, 1344
 \@gls@preglossaryhook
 225, 227, 586, 587, 838,
 839, 1094, 1095, 1250, 1251, 1407
 \@gls@prevlevel 333, 334,
 416–419, 678, 679, 704, 705,
 716, 717, 931, 932, 957, 958,
 970, 971, 1448, 1449, 1453, 1454
 \@gls@provide@glossarygroup 263, 264
 \@gls@provide@newglossary
 79, 81, 505, 755, 756,
 1010–1012, 1165, 1166, 1322, 1323
 \@gls@quotechar
 . 137–140, 193, 195, 196, 198,
 540–544, 569–572, 671, 792–
 795, 820–824, 924, 1045–1049,
 1075, 1077, 1079, 1200–1203,
 1231–1235, 1357–1360, 1388–1392
 \@gls@reference . . . 206, 209, 250,
 577, 578, 580, 595, 829, 830,
 832, 847, 1085, 1087, 1102,
 1240, 1243, 1258, 1397, 1399, 1422
 \@gls@removedoption 30, 35,
 42, 984, 988, 992, 1139, 1142,
 1143, 1146, 1147, 1294, 1298, 1302
 \@gls@removespaces 267, 268, 602,
 854, 1108, 1109, 1264, 1265, 1430
 \@gls@renewglossary
 199, 213, 573, 581, 825, 833,
 1080, 1088, 1235, 1244, 1392, 1401

\@gls@replacementtext	132, 537, 788, 1042, 1197, 1354
..... 437, 438, 637, 890, 891	
\@gls@requested@languages	\@gls@setsort 18–
..... 34, 34, 52,	20, 132, 186, 474, 475, 538,
987, 996, 1142, 1151, 1297, 1307	564, 723–725, 788, 816, 978,
\@gls@rest 177,	979, 1042, 1071, 1132–1134,
560, 811, 812, 1066, 1222, 1379	1197, 1226, 1287, 1288, 1354, 1383
\@gls@restore@glsllocal 129, 133,	\@gls@setupshortcuts 49,
1040, 1042, 1195, 1197, 1352, 1354	488, 489, 739, 994, 995, 1149, 1306
\@gls@restoreat	\@gls@setupsort@clear
.. 93, 512, 763, 1019, 1173, 1330 20, 979, 1134, 1288
\@gls@roman . 65, 66, 497, 498, 669,	\@gls@setupsort@def
748, 922, 1003, 1004, 1158, 1315	... 18, 474, 724, 978, 1133, 1287
\@gls@run@output@dir	\@gls@setupsort@none
. 41, 42, 46, 48, 1301, 1304, 1305	... 20, 475, 725, 979, 1133, 1288
\@gls@run@output@file ... 46, 48,	\@gls@setupsort@standard ... 17,
200–202, 1304, 1305, 1393–1395	473, 474, 723, 977, 978, 1132, 1287
\@gls@run@unrestricted@shell 46,	\@gls@setupsort@use
47, 200, 201, 203, 1304, 1393–1395	... 19, 474, 724, 978, 1133, 1287
\@gls@runshell ... 46, 46, 47, 48,	\@gls@sort
200, 202, 1304, 1305, 1393, 1395	235, 236, 245, 248, 594, 846,
\@gls@sanitized@tmp 135,	1101, 1257, 1412, 1413, 1419, 1421
539, 540, 790, 1044, 1199, 1356	\@gls@sort@A 238, 239, 590,
\@gls@sanitizedesc	842, 1097, 1098, 1253, 1414, 1415
..... 26, 478, 483, 728, 733	\@gls@sort@B 238, 239, 590,
\@gls@sanitizename	842, 1097, 1098, 1253, 1414, 1415
... 26, 478, 728, 982, 1137, 1292	\@gls@startswithexpandonce
\@gls@sanitizesort	90, 91, 511, 761, 1017, 1172, 1329
.. 17, 27, 474, 478, 723, 728,	\@gls@storenonumberlist ... 87,
977, 982, 1132, 1137, 1287, 1292	107, 508, 521, 758, 759, 772,
\@gls@sanitizesymbol ... 27, 478,	1015, 1028, 1169, 1183, 1326, 1340
483, 484, 728, 733, 982, 1137, 1292	\@gls@symbolsdef
\@gls@save@glsllocal 128, 129, 132,	44, 44, 487, 737, 993, 1147, 1303
1040, 1042, 1195, 1197, 1352, 1354	\@gls@this
\@gls@saveentrycounter 132, 133,	. 214, 582, 834, 1089, 1244, 1401
186, 538, 564, 788, 789, 816,	\@gls@thisHloc
1042, 1071, 1197, 1226, 1354, 1383	. 220, 584, 836, 1092, 1247, 1404
\@gls@savenonumberlist ... 86, 87,	\@gls@thisentryname
87, 208, 508, 758, 759, 1014,	.. 39, 735, 990, 1144, 1145, 1300
1015, 1087, 1169, 1242, 1326, 1399	\@gls@thisfield
\@gls@see@noindex .. 9, 9, 10, 86,	.. 78, 504, 754, 1010, 1165, 1322
470, 471, 508, 720, 758, 974,	\@gls@thislabel 75, 222,
1014, 1129, 1169, 1283, 1284, 1326	241, 502, 585, 592, 753, 837,
\@gls@setacrstyle .. 34, 49, 483,	844, 1008, 1093, 1099, 1163,
484, 488, 489, 733, 739, 987,	1248, 1249, 1255, 1320, 1405, 1416
994, 995, 1142, 1149, 1298, 1306	\@gls@thislang
\@gls@setcounter ... 81, 82, 506, 52, 996, 1151, 1307, 1308
756, 1012, 1166, 1167, 1323, 1324	\@gls@thislist 184, 185,
\@gls@setdefault@glslink@opts .	564, 815, 816, 1070, 1226, 1383
..... 131,	\@gls@thisloc
	. 220, 584, 836, 1092, 1247, 1404

<code>\@gls@thisval</code>	<code>\@gls@writeglossarysubgroup</code>
.. 95, 513, 764, 1020, 1175, 1332 40, 263, 263
<code>\@gls@title</code> . 493, 743, 999, 1153, 1310	<code>\@gls@writeisthook</code> 193,
<code>\@gls@tmp</code> 9, 53, 66, 67, 91,	195, 195, 569, 570, 820, 822,
135, 213, 267, 470, 475, 490,	1075–1077, 1231, 1232, 1388, 1389
498, 511, 539, 540, 581, 688,	<code>\@gls@x</code> 267, 854, 1109, 1264, 1265, 1430
689, 719, 724, 741, 748, 749,	<code>\@gls@xdy@Lclass@Alpha-page-numbers</code>
761, 790, 833, 942, 974, 979, 66
996, 1003, 1004, 1017, 1044,	<code>\@gls@xdy@Lclass@Appendix-page-numbers</code>
1088, 1109, 1125, 1126, 1128, 66
1133, 1151, 1158, 1159, 1172,	<code>\@gls@xdy@Lclass@Roman-page-numbers</code>
1199, 1244, 1264, 1283, 1308, 66
1315, 1316, 1329, 1356, 1401, 1430	<code>\@gls@xdy@Lclass@alpha-page-numbers</code>
<code>\@gls@tmpb</code> 136, 136, 137– 66
142, 196–198, 540–545, 570–	<code>\@gls@xdy@Lclass@arabic-page-numbers</code>
572, 791–797, 822–824, 1045– 66
1050, 1077, 1079, 1200–1205,	<code>\@gls@xdy@Lclass@arabic-section-numbers</code>
1233–1235, 1357–1362, 1390–1392 66
<code>\@gls@toc</code> . . 60, 61, 494, 495, 744,	<code>\@gls@xdy@Lclass@roman-page-numbers</code>
745, 1000, 1154, 1155, 1311, 1312 65
<code>\@gls@type</code> 202, 205, 276,	<code>\@gls@xdy@locationlist</code>
416, 576, 577, 606, 616, 618, 65, 190, 497,
620, 621, 623, 626, 627, 629,	566, 567, 747, 818, 1003, 1073,
715, 828, 829, 858, 868, 871,	1158, 1228, 1229, 1315, 1385, 1386
872, 874, 876, 879, 880, 882,	<code>\@gls@xdycheckbackslash</code> . . 135,
969, 1082, 1084, 1113, 1238,	142, 539, 545, 790, 796, 797,
1239, 1269, 1395, 1396, 1434, 1452	1044, 1050, 1199, 1205, 1356, 1362
<code>\@gls@updatechecked</code> 135,	<code>\@gls@xdycheckquote</code>
136, 136, 196, 197, 539, 540, 135, 141, 539, 544,
571, 572, 790, 791, 823, 824,	790, 796, 1044, 1049, 1050,
1044, 1045, 1078, 1199, 1200,	1199, 1204, 1205, 1356, 1361, 1362
1233, 1234, 1356, 1357, 1390, 1391	<code>\@gls@xref</code> 221,
<code>\@gls@usetranslator</code>	585, 836, 837, 1092, 1248, 1405
32, 33, 34, 52, 482, 483, 490,	<code>\@gls@Alphacompositor</code> . 56, 56, 66,
731, 732, 740, 986, 987, 996,	492, 498, 669, 742, 748, 923,
1140–1142, 1151, 1296, 1297, 1308	998, 1004, 1153, 1159, 1310, 1316
<code>\@gls@value</code> 89, 90, 180, 510,	<code>\@gls@Hlocref</code>
511, 561, 760, 761, 813, 1016,	216, 219, 582, 583, 834, 835,
1017, 1068, 1171, 1223, 1328, 1380	1090, 1091, 1245, 1246, 1402, 1403
<code>\@gls@warnonglossdefined</code> . . . 25,	<code>\@gls@sacronymlists</code> 22, 23,
26, 224, 478, 586, 728, 838,	24, 71, 274, 276, 298, 476, 477,
982, 1093, 1137, 1249, 1291, 1406	501, 605, 606, 616, 618–621,
<code>\@gls@warnontheGLOSSdefined</code> 25,	623, 625–627, 629, 631, 726,
26, 257, 478, 598, 728, 850,	751, 857, 858, 868, 870–872,
982, 1105, 1137, 1261, 1291, 1426	874, 876, 878–880, 882, 884,
<code>\@gls@write@entrycounts</code> 118, 119,	980, 981, 1007, 1112, 1113,
527, 528, 778, 1035, 1190, 1347	1124, 1135, 1162, 1267–1269,
<code>\@gls@writedef</code> 93, 94, 512, 762, 763,	1280, 1290, 1319, 1433, 1434, 1445
1018, 1019, 1173, 1174, 1330, 1331	<code>\@gls@saddkey</code>
<code>\@gls@writeglossarygroup</code> . . . 40, 263	96, 97, 514, 765, 1021, 1176, 1333

<code>\@glsaddstoragekey</code>	96, 514, 764, 765, 1020, 1021, 1175, 1332
<code>\@glsaddxdyattribute</code> 63, 64, 496, 746, 1002, 1156, 1157, 1313, 1314	
<code>\@glsdefaultplural</code>	89, 510, 760
<code>\@glsdefaultsort</code>	
.. 17, 89, 474, 510, 723, 760, 977, 1016, 1132, 1171, 1287, 1328	
<code>\@glsdesc</code>	
. 156, 551, 803, 1056, 1212, 1369	
<code>\@glsdesc@</code>	156, 451, 551, 650, 803, 903, 1056, 1212, 1369
<code>\@glsdescplural</code>	157, 551, 803, 1056, 1057, 1212, 1369
<code>\@glsdescplural@</code>	157, 451, 551, 650, 803, 903, 1057, 1212, 1369
<code>\@glsdisp</code>	150, 150, 151, 549, 800, 1054, 1209, 1366
<code>\@glsentry</code>	39, 115, 116, 119, 485, 525, 526, 528, 735, 776– 779, 990, 1033–1036, 1144, 1145, 1188, 1190, 1300, 1345, 1347
<code>\@glsentrytitlecase</code> . . .	180, 180, 561, 562, 813, 1068, 1223, 1380
<code>\@glsfirst</code>	152, 550, 801, 1054, 1055, 1210, 1367
<code>\@glsfirst@</code>	152, 450, 550, 649, 801, 902, 1055, 1210, 1367
<code>\@glsfirstletter</code>	188, 188, 565, 817, 1071, 1072, 1227, 1384
<code>\@glsfirstplural</code>	
. 154, 550, 802, 1055, 1211, 1368	
<code>\@glsfirstplural@</code>	
. 154, 450, 450, 550, 649, 802, 902, 1055, 1211, 1368	
<code>\@glsnumber</code>	266, 266, 602, 853, 854, 1108, 1263, 1264, 1429
<code>\@glsisacronymlistfalse</code>	
... 23, 477, 726, 981, 1135, 1290	
<code>\@glsisacronymlisttrue</code>	
... 23, 477, 726, 981, 1135, 1290	
<code>\@glslink</code>	
112, 113, 132, 144, 144, 145, 185, 300, 538, 545, 546, 564, 688, 789, 797, 798, 816, 941, 1031, 1032, 1042, 1051, 1071, 1125, 1186, 1197, 1206, 1207, 1226, 1343, 1354, 1363, 1364, 1383	
<code>\@glslocalreset</code>	114, 115, 117, 525, 526, 775–777, 1032–1034, 1187–1189, 1344–1346
<code>\@glslocalunset</code>	114, 114, 117, 525, 526, 776, 777, 1033, 1034, 1188, 1189, 1345, 1346
<code>\@glslocref</code>	209, 216, 219, 220, 580, 582–584, 668, 832, 834– 836, 921, 1087, 1090, 1091, 1243, 1245–1247, 1400, 1402–1404
<code>\@glsmakeindexappname</code>	55, 204
<code>\@glsminrange</code>	
. 189, 189, 190, 566, 567, 669, 670, 817–819, 922, 923, 1072, 1073, 1228, 1229, 1385, 1386	
<code>\@glsname</code>	
. 155, 551, 802, 1056, 1211, 1368	
<code>\@glsname@</code>	155, 451, 551, 649, 802, 903, 1056, 1211, 1368
<code>\@glsnavhypertarget</code>	
. 301, 302, 688, 941, 1125	
<code>\@glsnextpages</code>	
226, 255, 587, 597, 839, 848, 1095, 1104, 1250, 1260, 1407, 1425	
<code>\@glsnodesc</code>	88, 103, 105, 107, 509, 519, 522, 760, 769, 770, 772, 1016, 1025, 1026, 1028, 1029, 1170, 1180, 1181, 1183, 1327, 1337, 1338, 1340
<code>\@glsnoname</code>	
. 88, 103, 106, 107, 509, 519– 521, 759, 769, 771, 772, 1015, 1025, 1027, 1028, 1170, 1180, 1182, 1183, 1327, 1337, 1339, 1340	
<code>\@glsnonextpages</code>	
226, 255, 587, 596, 839, 848, 1095, 1104, 1250, 1259, 1407, 1425	
<code>\@glsnumberformat</code>	128, 132, 185, 186, 209, 219, 220, 536, 537, 564, 580, 584, 668, 787, 788, 816, 832, 835, 836, 921, 1040, 1042, 1071, 1087, 1091, 1195, 1197, 1226, 1243, 1247, 1352, 1354, 1383, 1400, 1403, 1404
<code>\@glsopenfile</code>	199, 199, 212, 573, 581, 825, 832, 1080, 1088, 1235, 1236, 1243, 1392, 1393, 1400
<code>\@glsorder</code>	35, 203, 484, 576, 733, 828, 988, 1083, 1143, 1239, 1298, 1395, 1396
<code>\@glspl</code>	147, 148, 547, 799, 1052, 1208, 1365

<code>\@glspl@</code>	118, 120,	<code>\@glseriii</code>	162,
	148, <u>148</u> , 425, 426, 527, 528,		553, 805, 1058, 1059, 1214, 1371
	547, 685–687, 778, 779, 799,	<code>\@glseriii@</code>	162, <u>453</u> , 553,
	939, 940, 1035, 1036, 1052,		651, 805, 904, 1059, 1214, 1371
	1190, 1191, 1208, 1347, 1348, 1365	<code>\@glseriv</code>	163, 553,
<code>\@glsplural</code>	153, 550,		805, 1059, 1214, 1215, 1371, 1372
	802, 1055, 1210, 1211, 1367, 1368	<code>\@glseriv@</code>	163, <u>453</u> , 554,
<code>\@glsplural@</code>	153, <u>450</u> , 550,		651, 805, 904, 1059, 1215, 1372
	649, 802, 902, 1055, 1211, 1368	<code>\@glserv</code>	
<code>\@glsprefix@record@hook</code>			164, 554, 805, 1059, 1215, 1372
	. <u>423</u> , 424–427, 685–687, 938–941	<code>\@glserv@</code>	164, <u>454</u> , 554,
<code>\@glreset</code>	113,		651, 806, 905, 1059, 1215, 1372
	<u>115</u> , 117, 524–526, 775–777,	<code>\@glservi</code>	
	1032–1034, 1187–1189, 1344–1346		165, 554, 806, 1060, 1215, 1372
<code>\@glsee</code> 109, 222, 522, 585, 773, 837,		<code>\@glservi@</code>	165, <u>454</u> , 554,
	1029, 1093, 1184, 1248, 1341, 1405		652, 806, 905, 1060, 1215, 1372
<code>\@glsshowacccsupp</code> . . . 7, 8, <u>9</u> , 429,		<code>\@glswidestname</code>	366, 375,
	469, 470, 632, 719, 720, 885,		416, <u>416</u> , 417, 418, 678, 679,
	973, 974, 1127, 1128, 1282, 1283		716, 717, 932, 969, 970, 1453, 1454
<code>\@glsshowtarget</code>	7, 8, <u>8</u> ,	<code>\@glswritefiles</code>	
	469, 470, 545, 718, 719, 797,		42, <u>211</u> , 486, 580, 736, 832,
	972–974, 1127, 1128, 1282, 1283		991, 1087, 1146, 1243, 1302, 1400
<code>\@glssymbol</code>	158, 552,	<code>\@glxtr@doacccsupp</code>	428, 631, 885
	803, 1057, 1212, 1213, 1369, 1370	<code>\@glxtr@record</code>	424, 685, 938
<code>\@glssymbol@</code>	158, <u>452</u> , 552,	<code>\@gobble</code> 7, 18–20, 38, 39, 94,	
	650, 803, 903, 1057, 1213, 1370		112, 135, 187, 188, 191, 204,
<code>\@glssymbolplural</code>			206, 306, 469, 474, 475, 484,
	. 159, 552, 804, 1057, 1213, 1370		485, 512, 513, 524, 539, 565,
<code>\@glssymbolplural@</code> 159, <u>452</u> , 552,			567, 568, 576, 577, 667, 670,
	650, 804, 903, 1057, 1213, 1370		671, 718, 719, 723–725, 734,
<code>\@glstarget</code>	144,		735, 763, 764, 775, 790, 816,
	<u>144</u> , 145, 258, 301, 546, 598,		817, 819, 828, 829, 920, 923,
	688, 797, 798, 850, 941, 1051,		924, 944, 972, 973, 978, 979,
	1105, 1125, 1207, 1261, 1364, 1426		989, 990, 1019, 1020, 1031,
<code>\@glstext</code>			1044, 1071, 1074, 1083, 1084,
	. 151, 549, 801, 1054, 1210, 1367		1127, 1132, 1133, 1144, 1145,
<code>\@glstext@</code>	151, <u>449</u> , 549,		1174, 1186, 1199, 1227, 1229,
	649, 801, 902, 1054, 1210, 1367		1230, 1239, 1240, 1282, 1287,
<code>\@glunset</code>	114,		1288, 1299, 1300, 1331, 1343,
	<u>114</u> , 117, 525, 526, 776, 777,		1356, 1384, 1386, 1387, 1396, 1397
	1033, 1034, 1187–1189, 1344–1346	<code>\@gobblethree</code>	
<code>\@glseri</code>			7, 469, 719, 973, 1127, 1282
	. 160, 552, 804, 1058, 1213, 1370	<code>\@idxitem</code> . . 410, 712, 966, 1449, 1450	
<code>\@glseri@</code>	160, <u>452</u> , 552,	<code>\@ifclassloaded</code>	
	651, 804, 904, 1058, 1213, 1370		6, 15, 59, 468, 472, 493,
<code>\@glserii</code>			718, 722, 743, 972, 976, 999,
	. 161, 553, 804, 1058, 1214, 1371		1127, 1131, 1154, 1281, 1285, 1311
<code>\@glserii@</code>	161, <u>453</u> , 553,	<code>\@ifglossaryexists</code>	71, <u>72</u> ,
	651, 804, 904, 1058, 1214, 1371		224, 501, 586, 751, 752, 838,
			1007, 1093, 1162, 1249, 1319, 1406

<code>\@ifl@t@r</code>	29, 480, 730, 984, 1139, 1294	<code>\@newglossary</code>	79, 81, 82, 505, 506, 755, 756, 1011, 1012, 1165–1167, 1322–1324
<code>\@ifnextchar</code>	81, 129, 506, 536, 756, 787, 1012, 1041, 1166, 1195, 1323, 1352	<code>\@newglossaryentry@defcounters</code>	108, 109, 116, 522, 523, 526, 773, 777, 1029, 1034, 1184, 1189, 1341, 1346
<code>\@ifpackageloaded</code>	6, 12, 14, 32–34, 52, 70, 112, 184, 195, 428, 468, 471, 482, 483, 490, 500, 524, 563, 570, 631, 718, 721, 731–733, 740, 751, 775, 815, 822, 884, 972, 975, 976, 986, 987, 996, 1006, 1031, 1070, 1077, 1127, 1130, 1140– 1142, 1151, 1161, 1186, 1225, 1232, 1281, 1284, 1285, 1296, 1297, 1308, 1318, 1343, 1382, 1389	<code>\@newglossaryentryposthook</code>	96, 97, 108, 109, 422, 433, 514, 522, 523, 634, 684, 765, 773, 888, 937, 1021, 1029, 1175, 1176, 1184, 1332, 1333, 1341
<code>\@ifstar</code>	71, 80, 96, 129, 188, 269, 501, 505, 514, 536, 565, 603, 751, 755, 764, 765, 787, 817, 855, 1007, 1011, 1020, 1021, 1041, 1071, 1072, 1109, 1162, 1166, 1175, 1176, 1195, 1227, 1265, 1319, 1323, 1332, 1333, 1352, 1384, 1430	<code>\@newglossaryentryprehook</code>	96, 97, 102, 104, 109, 421, 432, 514, 518, 519, 523, 634, 683, 765, 769, 770, 773, 887, 937, 1021, 1025, 1026, 1029, 1175, 1176, 1180, 1181, 1184, 1332, 1333, 1337, 1338, 1341
<code>\@ifundefined</code>	52, 310, 321, 335, 342, 417, 418, 490, 679, 688, 692, 698, 705, 709, 716, 717, 740, 932, 941, 946, 952, 958, 959, 962, 963, 970, 996, 1150, 1307, 1453, 1454	<code>\@nil</code>	23, 109, 134–136, 177, 196, 197, 219, 221, 236, 247, 266– 268, 477, 522, 539, 540, 560, 571, 572, 584, 585, 593, 594, 602, 668, 726, 773, 790, 791, 811, 823, 824, 835, 837, 845, 846, 853, 854, 921, 981, 1029, 1043–1045, 1066, 1078, 1091, 1092, 1100, 1101, 1108, 1109, 1135, 1184, 1198–1200, 1221, 1233, 1234, 1247, 1248, 1256, 1257, 1263–1265, 1290, 1341, 1355–1357, 1378, 1390, 1391, 1403, 1405, 1413, 1420, 1429, 1430
<code>\@ignored@glossaries</code>	39, 82, 83, 83, 485, 506, 507, 735, 756, 757, 990, 1012, 1013, 1144, 1167, 1300, 1324	<code>\@nnil</code>	23, 222, 477, 585, 726, 837, 981, 1093, 1135, 1248, 1290, 1405
<code>\@input@</code>	227, 588, 839, 1095, 1251, 1408	<code>\@no@makeglossaries</code>	38, 204, 207, 209, 484, 485, 577, 578, 580, 734, 828, 830, 832, 989, 1084, 1085, 1087, 1143, 1144, 1239, 1241, 1242, 1299, 1396, 1398, 1399
<code>\@istfilename</code>	55, 203, 492, 576, 742, 828, 998, 1083, 1152, 1239, 1309, 1395, 1396	<code>\@no@post@desc</code>	54, 76, 492, 673, 742, 926, 997, 1152, 1309
<code>\@makecol</code>	320, 697, 698, 951	<code>\@nopostdesc</code>	54, 226, 491, 587, 742, 839, 997, 1095, 1152, 1250, 1309, 1407
<code>\@makeglossaries@warn@noprintglossary</code>	204, 205, 828, 829, 1083, 1084, 1239, 1240, 1396	<code>\@onelevel@sanitize</code>	9, 27, 39, 65, 66, 94, 111, 135, 188, 192, 221, 223, 247, 267, 470, 479, 498, 512, 524, 539, 540, 565, 566, 568, 585,
<code>\@makeglossary</code>	38, 199, 204, 484, 485, 573, 576, 734, 735, 825, 828, 989, 1080, 1083, 1144, 1235, 1239, 1299, 1392, 1396		
<code>\@minus</code>	306, 329, 378, 410, 691, 701, 712, 944, 955, 966, 1446, 1449		
<code>\@mkboth</code>	59, 493, 494, 743, 744, 999, 1154, 1311		
<code>\@ne</code>	112, 1031, 1186, 1343		

586, 593, 669, 670, 719, 728,
729, 735, 748, 763, 774, 790,
817, 820, 836, 838, 845, 922,
924, 974, 983, 990, 1003, 1019,
1030, 1044, 1072, 1074, 1092,
1093, 1100, 1109, 1128, 1137,
1145, 1158, 1174, 1185, 1199,
1227, 1228, 1230, 1248, 1249,
1256, 1264, 1283, 1292, 1300,
1315, 1331, 1342, 1356, 1384,
1385, 1387, 1405, 1406, 1420, 1430
\@onlypreamble 81, 92, 103,
118, 122, 205, 209, 506, 512,
518, 527, 529, 577, 579, 756,
762, 769, 778, 780, 829, 831,
1012, 1018, 1025, 1035, 1037,
1084, 1087, 1167, 1172, 1180,
1190, 1192, 1239, 1242, 1324,
1329, 1337, 1347, 1349, 1396, 1399
\@onlypremakeg . . . 50, 55, 56, 63,
64, 67, 82, 195, 489, 492, 496,
499, 506, 570, 739, 742, 746,
747, 749, 756, 822, 995, 998,
1001, 1002, 1005, 1012, 1077,
1150, 1152, 1153, 1156, 1157,
1159, 1167, 1232, 1307, 1309,
1310, 1313, 1314, 1316, 1324, 1389
\@org@Gls@acrentryname
. 176, 1066, 1221, 1378
\@org@glossaryentrynumbers
. 226, 227, 587, 839,
1094, 1095, 1250, 1251, 1407, 1408
\@org@Gls@assign@descplural . . .
. 616, 623,
625, 627, 661, 664, 665, 868,
875, 876, 878, 880, 915, 917, 918
\@org@Gls@assign@firstpl
. 616–619, 621,
623, 625–627, 661–665, 867,
868, 870–876, 878, 880, 914–918
\@org@Gls@assign@plural 616–619,
621, 623, 625–627, 661–665,
868, 870–876, 878, 880, 915–918
\@org@Gls@assign@symbolplural .
. 616–619, 621,
625–627, 662, 663, 665, 868,
870–874, 878, 880, 915, 916, 918
\@org@Gls@numberformat
. 184, 563, 815, 1070, 1225, 1382
\@org@newglossaryentryprehook .
. 102, 518, 769, 1025, 1180, 1337
\@outputpage 320, 697, 698, 951
\@p@glossarysection
. . 58, 60, 493, 494, 743, 744,
999, 1000, 1153, 1154, 1310, 1311
\@pgls 424, 424, 685, 938
\@pgls@ 424, 424, 685, 938
\@pglspl 424, 424, 685, 938
\@pglspl@ 424, 425, 685, 938
\@plus . . . 306, 329, 378, 410, 691,
701, 712, 944, 955, 966, 1446, 1449
\@print@glossary
224, 227, 586, 588, 838, 839,
1094, 1095, 1249, 1251, 1406, 1408
\@print@noidx@glossary
225, 242, 586, 592, 838, 844,
1094, 1100, 1250, 1255, 1406, 1417
\@printgloss@checkexists . . 224,
224, 225, 226, 586, 587, 838,
839, 1094, 1249, 1250, 1406, 1407
\@printgloss@checkexists@allowignored
. 224, 586, 838, 1094, 1249, 1406
\@printgloss@checkexists@noignored
. 224, 224, 225, 586,
838, 1093, 1094, 1249, 1250, 1406
\@printgloss@setsort
. 204, 207, 225, 226,
577, 578, 586, 587, 828, 830,
838, 839, 1084, 1085, 1094,
1239, 1241, 1250, 1396, 1398, 1407
\@printglossary 224, 225, 225, 586,
838, 1094, 1249, 1250, 1406, 1407
\@provide@glossary@seclabel . 10, 11
\@roman 65, 497,
669, 748, 922, 1003, 1158, 1315
\@secondofthree 129, 145,
146, 149, 167, 169, 171, 173,
447, 536, 546–548, 555–557,
559, 647, 787, 798, 800, 807–
810, 900, 1041, 1051–1053,
1061–1064, 1195, 1207, 1209,
1216, 1217, 1219, 1220, 1352,
1364, 1366, 1373, 1374, 1376, 1377
\@secondoftwo 29, 32–34,
53, 94, 95, 112, 144–147, 150,
166–168, 170–172, 216, 306,
419, 420, 447–449, 479, 480,
482, 483, 490, 512, 513, 546,
547, 549, 555, 557, 558, 647,

648, 666, 683, 729, 731–733,	\@wrglossary@pageformat .. 215,
740, 763, 764, 797–800, 806,	215, 582, 834, 1089, 1245, 1402
807, 809, 900–902, 919, 936,	\@wrglossarynumberhook ... 215,
944, 983, 984, 986, 987, 996,	215, 219, 582, 583, 834, 835,
1019, 1020, 1031, 1032, 1051,	1089, 1091, 1245, 1246, 1402, 1403
1052, 1054, 1060, 1061, 1063,	\@write@glslabelsname
1090, 1138, 1140–1142, 1151, 39, 39, 735, 990, 1145, 1300
1174, 1175, 1186, 1207–1209,	\@xdy@main@language
1216–1219, 1245, 1293, 1294,	.. 36, 37, 484, 574, 575, 588,
1296, 1297, 1308, 1331, 1332,	734, 826, 827, 840, 988, 1143, 1298
1343, 1364–1366, 1373–1376, 1402	\@xdyattributelist
\@set@glo@numformat ... 134, 220,	62,
539, 584, 668, 790, 836, 921,	64, 191, 495, 496, 567, 745,
1043, 1091, 1198, 1247, 1355, 1404	746, 819, 1001, 1002, 1074,
\@sglsaddkey	1156, 1157, 1229, 1312, 1314, 1386
96, 97, 514, 765, 1021, 1176, 1333	\@xdyattributes
\@sglsaddstoragekey 62, 63, 64, 190, 495, 496,
.. 96, 514, 764, 1020, 1175, 1332	566, 667, 669, 745, 746, 818,
\@tabacckludge	920, 922, 1001, 1002, 1073,
... 29, 479, 729, 983, 1138, 1293	1155–1157, 1228, 1312–1314, 1385
\@text@composite@x	\@xdycounters
... 29, 479, 729, 983, 1138, 1293	62,
\@thirdofthree 130, 147, 149, 168,	64, 191, 495, 496, 567, 745,
170, 172, 174, 448, 536, 547,	746, 819, 1001, 1002, 1074,
548, 555, 557–559, 647, 787,	1156, 1157, 1229, 1313, 1314, 1386
799, 800, 807–809, 811, 901,	\@xdycrossrefhook
1041, 1052, 1053, 1061–1063,	.. 67, 191, 498, 567, 749, 819,
1065, 1196, 1208, 1209, 1217–	1004, 1074, 1159, 1229, 1316, 1386
1220, 1353, 1365, 1366, 1374–1377	\@xdy@language
\@this@attr 191, 567, 568,	.. 69, 228, 500, 588, 750, 840,
819, 1074, 1229, 1230, 1386, 1387	1006, 1095, 1161, 1251, 1318, 1408
\@this@childlabel	\@xdylettergroups 70, 70, 193, 501,
.. 237, 589, 841, 1096, 1252, 1413	569, 671, 751, 820, 924, 1006,
\@this@counter	1007, 1075, 1161, 1231, 1318, 1388
.. 64, 496, 746, 1002, 1157, 1314	\@xdylocationclassorder
\@this@ctr 68, 68, 191,
.. 191, 567, 819, 1074, 1229, 1386	499, 567, 670, 749, 819, 923,
\@this@key	1005, 1074, 1160, 1229, 1317, 1386
.. 95, 513, 764, 1020, 1175, 1332	\@xdylocoref
\@this@label 236, 237,	62, 63,
589, 840, 841, 1096, 1252, 1413	193, 495, 496, 568, 667, 671,
\@this@scs	745, 746, 820, 920, 924, 1001,
50, 489, 490, 740, 995, 1150, 1307	1002, 1075, 1156, 1230, 1313, 1387
\@tmp 498, 669, 922	\@xdynumbergrouporder
\@use@ption 70, 188, 188, 501,
... 46, 488, 738, 994, 1149, 1304	565, 566, 751, 817, 1007, 1072,
\@warn@nomakeglossaries	1161, 1227, 1228, 1318, 1384, 1385
202, 228, 576, 588, 827, 840,	\@xdyrequiredstyles .. 68, 69, 189,
1082, 1096, 1238, 1251, 1395, 1408	500, 566, 668, 750, 818, 922,
	1006, 1072, 1160, 1228, 1317, 1385
	\@xdysortrules 68, 68, 193,
	499, 569, 671, 750, 820, 924,
	1005, 1075, 1160, 1231, 1317, 1388

<code>\@xdystyle</code> . . . 189, 566, 668, 669, 818, 922, 1072, 1073, 1228, 1385	<code>\ac</code> . . . 288, 615, 867, 1122, 1277, 1443
<code>\@xdyuseralphabets</code> . . 65, 65, 190, 497, 566, 669, 747, 818, 922, 1003, 1073, 1158, 1228, 1315, 1385	<code>access (key)</code> 430
<code>\@xdyuserlocationdefs</code> . . . 67, 67, 191, 498, 499, 567, 667, 670, 748, 749, 819, 921, 923, 1004, 1005, 1073, 1159, 1229, 1316, 1386	<code>\accsuppglossaryentryfield</code> 455, 632, 653, 885, 906
<code>\@xdyuserlocationnames</code> 67, 67, 68, 498, 499, 668, 748, 749, 921, 1004, 1005, 1159, 1160, 1316, 1317	<code>\accsuppglossarysubentryfield</code> 455, 632, 653, 885, 906
<code>\@xfor@nextelement</code> 222, 585, 837, 1093, 1248, 1405	<code>\Acf</code> . . . 288, 615, 867, 1122, 1277, 1443
<code>\@xp@gl@getcounterprefix</code> 216, 219, 220, 1090, 1091, 1246, 1247, 1402–1404	<code>\acf</code> . . . 288, 615, 867, 1122, 1277, 1443
<code>\@xp@glshypernumber</code> 266, 266, 1108, 1263, 1429	<code>\Acfp</code> . . 288, 615, 867, 1122, 1277, 1443
<code>\@</code> 109, 110, 135, 141, 187, 194, 266, 267, 523, 539, 565, 569, 570, 602, 671, 672, 674, 675, 681, 682, 774, 790, 795, 796, 817, 821, 854, 924, 925, 927, 928, 934, 935, 1030, 1044, 1049, 1071, 1075, 1076, 1108, 1109, 1184, 1185, 1199, 1204, 1227, 1231, 1232, 1264, 1341, 1342, 1356, 1361, 1384, 1388, 1389, 1429, 1430	<code>\acfp</code> . . 288, 615, 867, 1122, 1277, 1443
<code>\^</code> 29, 39, 479, 729, 735, 983, 990, 1138, 1145, 1293, 1300	<code>\acl</code> . . . 288, 615, 867, 1122, 1277, 1443
<code>_</code> 210, 211, 229–236, 242–246, 250– 252, 328, 329, 352, 355–372, 374–396, 399, 402–410, 1409– 1412, 1417, 1418, 1420, 1422, 1423	<code>\aclp</code> . . 288, 615, 867, 1122, 1277, 1443
<code>\‘</code> 29, 479, 729, 983, 1138, 1293	<code>\aclp</code> . . 288, 615, 867, 1122, 1277, 1443
<code>\ </code> 136, 138, 141, 197, 540, 542, 572, 791, 793, 795, 823, 1045–1047, 1049, 1078, 1199, 1201, 1202, 1204, 1234, 1356, 1358, 1359, 1361, 1391	<code>\Acp</code> . . . 289, 615, 867, 1122, 1277, 1443
<code>\~</code> 29, 480, 729, 983, 1138, 1293	<code>\acp</code> . . . 288, 615, 867, 1122, 1277, 1443
A	<code>\acrfootnote</code> 289, 616, 617, 622, 869, 875, 1122, 1278, 1443
<code>\A</code> 262, 1106, 1262, 1427	<code>\ACRfull</code> 272, 604, 856, 1111, 1266, 1432
<code>\a</code> 29, 480, 729, 984, 1138, 1294	<code>\Acrfull</code> 272, 288, 604, 615, 855, 867, 1111, 1122, 1266, 1277, 1432, 1443
<code>\AA</code> 29, 480, 729, 984, 1138, 1294	<code>\acrfull</code> 272, 272, 288, 603, 615, 855, 867, 1110, 1111, 1122, 1266, 1277, 1431, 1432, 1443
<code>\aa</code> 29, 480, 729, 984, 1138, 1294	<code>\ACRfullfmt</code> 271, 273, 275, 283, 285, 460, 462, 604, 606, 612, 613, 657, 659, 856, 857, 864, 865, 910, 912, 1110– 1112, 1118, 1120, 1266–1268, 1274, 1276, 1431–1433, 1439, 1441
<code>\Ac</code> . . . 288, 615, 867, 1122, 1277, 1443	<code>\Acrfullfmt</code> 271, 272, 275, 283, 285, 460, 462, 604, 605, 612, 613, 657, 659, 856, 857, 864, 865, 910, 912, 1110– 1112, 1118, 1120, 1266, 1268, 1274, 1276, 1431–1433, 1439, 1441
	<code>\acrfullfmt</code> 271, 272, 275, 283, 285, 460, 462, 603, 605, 612, 613, 657, 658, 855, 857, 863, 865, 910, 912, 1110– 1112, 1118, 1120, 1266, 1268, 1274, 1275, 1431–1433, 1439, 1441
	<code>\acrfullformat</code> 183, 271, 271, 563, 603, 604, 615, 628, 815, 855, 867, 881, 1069, 1110, 1225, 1266, 1382, 1431
	<code>\ACRfullpl</code> 273, 604, 856, 1111, 1267, 1432

<code>\Acrfullpl</code>	605–615, 653–660, 857–867,
273, 288, 604, 615, 856, 867,	907–909, 911–914, 1112–1117,
1111, 1122, 1267, 1277, 1432, 1443	1119–1121, 1268, 1270–1273,
<code>\acrfullpl</code>	1275–1277, 1433–1438, 1440–1442
273, 288, 604, 615, 856, 867,	<code>\acronymfont</code>
1111, 1122, 1267, 1277, 1432, 1443	125,
<code>\ACRfullplfmt</code>	166–170, 177, 183, 274, 274,
272, 274,	276, 278–287, 444, 445, 447–
275, 284, 285, 461, 463, 605,	449, 456–465, 533, 534, 555–
606, 612, 613, 657, 659, 856–	557, 560, 563, 605–615, 617,
858, 864, 865, 910, 912, 1110–	618, 620–626, 645–648, 653–
1112, 1119, 1120, 1266–1268,	660, 662–664, 784, 785, 806–
1274, 1276, 1431–1433, 1440, 1441	808, 812, 815, 857–867, 869,
<code>\Acrfullplfmt</code>	871, 872, 874–877, 879, 898–
272, 273,	902, 907–915, 917, 1038, 1039,
275, 284, 285, 461, 463, 604,	1060–1062, 1066, 1069, 1111,
606, 612, 613, 657, 659, 856,	1113–1121, 1193, 1194, 1216–
858, 864, 865, 910, 912, 1110–	1218, 1222, 1225, 1267, 1268,
1112, 1119, 1120, 1266–1268,	1270–1277, 1350, 1351, 1373–
1274, 1276, 1431–1433, 1440, 1441	1375, 1379, 1382, 1432, 1434–1442
<code>\acrfullplfmt</code>	<code>acronymlists</code> (option)
272, 273,	24
275, 284, 285, 460, 463, 604,	<code>\acronymname</code>
606, 612, 613, 657, 659, 856,	22, 51, 54,
858, 864, 865, 910, 912, 1110–	475, 476, 490, 491, 725, 726,
1112, 1118, 1120, 1266–1268,	740, 741, 980, 995, 997, 1134,
1274, 1276, 1431–1433, 1440, 1441	1135, 1150, 1152, 1289, 1307, 1309
<code>\acrlinkfootnote</code>	<code>acronyms</code> (option)
289,	22
289, 616, 869, 1122, 1278, 1443	<code>\acronymsort</code>
<code>\acrlinkfullformat</code>	275, 276,
.	278–282, 284–287, 456–459,
271, 271, 272, 603–	461, 462, 464, 465, 605–615,
605, 855–857, 1110, 1266, 1431	653–655, 658, 660, 857–867,
<code>\ACRlong</code> 171, 558, 809, 1063, 1219, 1376	907–909, 911–914, 1112–1117,
<code>\Acrlong</code>	1119–1121, 1268, 1270–1273,
171,	1275, 1277, 1433–1438, 1440–1442
171, 288, 557, 615, 809, 867,	<code>\acronymtype</code>
1063, 1122, 1218, 1277, 1375, 1443	21, 22,
<code>\acrlong</code> 170, 288, 557, 615, 808, 867,	274–276, 475, 476, 605, 606,
1062, 1122, 1218, 1277, 1375, 1443	615–629, 661–664, 725, 726,
<code>\ACRlongpl</code>	857, 858, 867–882, 914–917,
. 173, 559, 810, 1064, 1220, 1377	979, 980, 1112, 1113, 1134,
<code>\Acrlongpl</code>	1135, 1267, 1268, 1289, 1433, 1434
173,	<code>\acrpluralsuffix</code> . 270, 275, 278–
173, 288, 558, 615, 810, 867,	280, 284–286, 456, 457, 461–
1064, 1122, 1220, 1277, 1377, 1443	464, 603, 605, 607–609, 612–
<code>\acrlongpl</code>	619, 621–623, 625, 626, 628,
172, 288, 558, 615, 810, 867,	653, 654, 658, 660–665, 855,
1064, 1122, 1219, 1277, 1376, 1443	857, 859–861, 864–867, 869–
<code>\acrnameformat</code> 274, 605, 620, 663,	871, 873–876, 878, 879, 881,
857, 873, 916, 1111, 1267, 1432	882, 907, 911–918, 1110, 1112,
<code>\acrno linkfootnote</code>	1114–1116, 1119–1121, 1265,
. 290, 616, 869, 1122, 1278, 1443	1268, 1270–1272, 1275, 1276,
<code>acronym</code> (option)	1431, 1433, 1435–1437, 1440–1442
21	<code>\ACRshort</code>
<code>\acronymentry</code> . 275, 276, 278–282,	
284–287, 456–459, 461–465,	

. 167 , 555 , 807 , 1061 , 1216 , 1373	\addtolength 418 , 679 , 717 , 932 , 970 , 1454
\Acrshort 167 , 167 , 287 , 555 , 615 , 807 , 867 , 1060 , 1121 , 1216 , 1277 , 1373 , 1442	\advance 105 , 112 , 133 , 320 , 474 , 475 , 520 , 538 , 697 , 724 , 771 , 789 , 951 , 978 , 979 , 1027 , 1031 , 1043 , 1133 , 1182 , 1186 , 1197 , 1339 , 1343 , 1354
\acrshort 166 , 287 , 555 , 615 , 806 , 867 , 1060 , 1121 , 1216 , 1277 , 1373 , 1442	\AE 29 , 480 , 729 , 984 , 1138 , 1294
\ACRshortpl 169 , 556 , 808 , 1062 , 1218 , 1375	\ae 29 , 480 , 729 , 984 , 1138 , 1294
\Acrshortpl 169 , 169 , 287 , 556 , 615 , 808 , 867 , 1062 , 1121 , 1217 , 1277 , 1374 , 1443	altlist (style) 308
\acrshortpl 168 , 287 , 556 , 615 , 807 , 867 , 1061 , 1121 , 1217 , 1277 , 1374 , 1443	altlistgroup (style) 308
acrstyles:	altlisthypergroup (style) 309
dua 282 , 459	altlong4col (style) 315
dua-desc 284 , 461	altlong4col-booktabs (style) 318
footnote 284 , 462	altlong4colborder (style) 316
footnote-desc 286 , 464	altlong4colheader (style) 315
footnote-sc 286 , 463	altlong4colheaderborder (style) 316
footnote-sc-desc 287 , 464	altlongragged4col (style) 325
footnote-sm 286 , 464	altlongragged4col-booktabs (style) 319
footnote-sm-desc 287 , 464	altlongragged4colborder (style) 326
long-sc-short 279	altlongragged4colheader (style) 325
long-sc-short-desc 281 , 457	altlongragged4colheaderborder (style) 326
long-short 277 , 456	\altnewglossary 81 , 506 , 756 , 1012 , 1167 , 1323
long-short-desc 280 , 457	altsuper4col (style) 340
long-sm-short 279	altsuper4colborder (style) 341
long-sm-short-desc 281 , 458	altsuper4colheader (style) 340
long-sp-short 278	altsuper4colheaderborder (style) 341
long-sp-short-desc 280	altsuperragged4col (style) 346
sc-short-long 280	altsuperragged4colborder (style) 347
sc-short-long-desc 282 , 458	altsuperragged4colheader (style) 347
short-long 279 , 456	altsuperragged4colheaderborder (style) 347
short-long-desc 281 , 458	alttree (style) 416
sm-short-long 280	alttreegroup (style) 418
sm-short-long-desc 282 , 459	alttreehypergroup (style) 419
\Acs 287 , 615 , 867 , 1121 , 1277 , 1442	\andname 52 , 223 , 490 , 585 , 740 , 837 , 996 , 1093 , 1150 , 1249 , 1307 , 1406
\acs 287 , 615 , 867 , 1121 , 1277 , 1442	\AnyTrackedLanguages 53 , 420 , 491 , 666 , 683 , 741 , 919 , 936 , 997 , 1151 , 1308
\Acsp 287 , 615 , 867 , 1121 , 1277 , 1443	\appto 24 , 30 , 87 , 96 , 97 , 195 , 216 , 421 , 422 , 432 , 477 , 480 , 509 , 514 , 634 , 683 , 684 , 727 , 730 , 759 , 765 , 887 , 936 , 937 , 981 , 984 , 1015 , 1021 , 1090 , 1136 , 1139 , 1170 , 1175 , 1176 , 1245 , 1290 , 1294 , 1327 , 1332 , 1333 , 1402
\acsp 287 , 615 , 867 , 1121 , 1277 , 1443	
\add@accent@ 29 , 479 , 729 , 983 , 1138 , 1293	
\addcontentsline 61 , 495 , 745 , 1001 , 1155 , 1312	
\addglossarytocaptions 52 , 53 , 490 , 740 , 741 , 996 , 1150 , 1151 , 1307 , 1308	

<code>\AtBeginDocument</code>	22,	327, 328, 356, 357, 359, 361,
	70, 93, 112, 186, 206, 476, 500,	365–368, 370–376, 389–393,
	512, 524, 564, 578, 725, 751,	402–405, 409, 1206, 1363, 1411
	763, 775, 816, 830, 980, 1006,	<code>\boolean</code>
	1019, 1031, 1071, 1085, 1134,	628, 881
	1161, 1173, 1186, 1226, 1240,	<code>\boolfalse</code>
	1289, 1318, 1330, 1343, 1383, 1397	42,
		43, 486, 736, 992, 1146, 1147, 1302
<code>\AtEndDocument</code>	38, 41, 93,	<code>\booltrue</code>
	118, 205, 208, 211, 227, 228,	486, 736
	302, 485, 486, 512, 527, 577,	<code>\bottomrule</code> . . .
	580, 588, 688, 735, 736, 763,	317, 318, 696, 949, 950
	778, 829, 832, 840, 941, 990,	<code>\box</code>
	991, 1019, 1035, 1084, 1087,	320, 697, 698, 951
	1095, 1125, 1144, 1146, 1173,	
	1190, 1240, 1243, 1251, 1300,	
	1301, 1330, 1347, 1397, 1400, 1408	
<code>automake</code> (option)	40	
<code>automakegloss</code> (option)	41	
<code>automakeglosslite</code> (option)	42	
B		
<code>\b</code>	29, 480, 729, 984, 1138, 1294	
<code>\begin</code> . . .	192, 243, 247, 307, 310–	
	313, 315, 316, 318–326, 329–	
	348, 568, 592, 670, 691–695,	
	697–712, 819, 844, 923, 944,	
	946–966, 1074, 1100, 1230,	
	1256, 1387, 1417, 1420, 1446–1449	
<code>\BeginAccSupp</code>	429, 632, 885	
<code>\begingroup</code>	7, 9, 39, 213,	
	218, 227, 267, 468, 470, 581,	
	583, 602, 718, 719, 735, 833,	
	835, 854, 972, 974, 990, 1088,	
	1090, 1108, 1127, 1128, 1145,	
	1244, 1246, 1251, 1264, 1281,	
	1283, 1300, 1401, 1403, 1408, 1429	
<code>\bfseries</code> .	311–315, 317, 318, 323,	
	324, 326, 336, 338–341, 343–	
	348, 693–696, 699–701, 706–	
	712, 946–950, 952–955, 959–966	
<code>\bgroup</code>	28,	
	102, 141, 184, 220, 237, 267,	
	479, 518, 563, 587, 589, 729,	
	769, 795, 815, 839, 841, 983,	
	1025, 1049, 1070, 1091, 1094,	
	1096, 1109, 1138, 1180, 1204,	
	1225, 1247, 1252, 1264, 1293,	
	1337, 1361, 1382, 1404, 1413, 1430	
<code>\bool</code>	27,	
	31, 72, 76, 143, 232–234, 251,	
		327, 328, 356, 357, 359, 361,
		365–368, 370–376, 389–393,
		402–405, 409, 1206, 1363, 1411
		<code>\boolean</code>
		628, 881
		<code>\boolfalse</code>
		42,
		43, 486, 736, 992, 1146, 1147, 1302
		<code>\booltrue</code>
		486, 736
		<code>\bottomrule</code> . . .
		317, 318, 696, 949, 950
		<code>\box</code>
		320, 697, 698, 951
C		
<code>\c</code>	28, 29, 210, 230, 352, 353,	
	355, 364, 365, 367, 368, 370–	
	378, 402, 403, 405, 406, 408,	
	480, 729, 984, 1138, 1292, 1293	
<code>\c@equation</code>		
	. 133, 538, 789, 1043, 1197, 1354	
<code>\c@glossaryentry</code>		
	. . . 15, 473, 722, 976, 1131, 1286	
<code>\c@glossarysubentry</code>		
	. . . 16, 473, 723, 977, 1132, 1286	
<code>\c@page</code>	214, 215,	
	218, 219, 582, 583, 834, 835,	
	1089–1091, 1245, 1246, 1401–1403	
<code>\capitalisewords</code>		
 180, 813, 1067, 1223, 1380	
<code>\catcode</code>	39, 93, 141, 512,	
	735, 763, 795, 990, 1019, 1049,	
	1145, 1173, 1204, 1300, 1330, 1361	
<code>\centering</code>	380–382	
<code>\cGls</code>	120,	
	120, 528, 779, 1036, 1191, 1348	
<code>\cgls</code>	119, 119, 120, 528,	
	779, 1036, 1190, 1191, 1347, 1348	
<code>\cGlsformat</code>		
	118, 120, 527, 528, 778, 779,	
	1035, 1036, 1190, 1191, 1347, 1348	
<code>\cglsformat</code>		
	117, 119, 527, 528, 778, 779,	
	1035, 1036, 1189, 1191, 1346, 1348	
<code>\cGlspl</code>	121, 121,	
	528, 529, 779, 1036, 1191, 1348	
<code>\cglspl</code>	120, 120,	
	121, 528, 779, 1036, 1191, 1348	
<code>\cGlsplformat</code>		
	118, 121, 527, 529, 778, 780,	
	1035, 1037, 1190, 1191, 1347, 1348	
<code>\cglsplformat</code>		
	118, 121, 527, 528, 778, 779,	
	1035, 1036, 1190, 1191, 1347, 1348	

<code>\changes</code>	184	<code>\csedef</code> 119, 215, 527, 582, 778, 834,	
<code>\char</code> ..	262, 600, 852, 1107, 1262, 1427	1035, 1089, 1190, 1245, 1347, 1402	
<code>\cleardoublepage</code>	61,	<code>\csgdef</code>	20, 58,
	494, 744, 745, 1000, 1155, 1312		79, 83, 117, 118, 223, 250, 251,
<code>\clearpage</code>	60,		493, 505, 506, 526, 527, 586,
	61, 494, 744, 745, 1000, 1155, 1312		595, 743, 755, 756, 777, 778,
<code>\closeout</code>	39, 93,		838, 847, 979, 999, 1011, 1012,
	193, 195, 200, 212, 485, 512,		1034, 1035, 1093, 1102, 1134,
	569, 570, 573, 581, 735, 763,		1153, 1165, 1167, 1189, 1190,
	820, 822, 825, 833, 990, 1019,		1249, 1258, 1288, 1310, 1322,
	1075, 1076, 1080, 1088, 1145,		1324, 1346, 1347, 1406, 1422, 1423
	1173, 1231, 1232, 1236, 1243,	<code>\cslet</code>	86, 87, 102,
	1300, 1330, 1388, 1389, 1393, 1400		109, 241, 509, 518, 523, 592,
<code>\compatglossarystyle</code>			759, 769, 774, 844, 1014, 1015,
	672–682, 926–936		1025, 1030, 1099, 1169, 1180,
<code>compatible-2.07</code> (option)	<u>42</u>		1184, 1255, 1326, 1337, 1341, 1416
<code>compatible-3.07</code> (option)	<u>42</u>	<code>\csname</code> .	17, 20, 29, 46, 51, 53, 54,
<code>\compatibleglossentry</code>	<u>258</u> ,		60, 63, 66, 69, 71, 75, 81, 82,
	<u>429</u> , 598, 599, 632, 850, 851, 885		89, 90, 96–98, 100, 101, 105–
<code>\compatiblesubglossentry</code>			112, 114, 115, 132–134, 146–
	260, <u>429</u> , 599, 632, 851, 885		150, 167–174, 184, 186, 190,
<code>\copy</code>	320, 321, 697, 698, 951		191, 197–200, 206, 209, 212,
<code>\count@</code>	247, 248, 593,		213, 215, 220, 221, 225–227,
	845, 1100, 1101, 1256, 1420, 1421		238, 264, 265, 269, 293–299,
<code>counter</code> (key)	<u>85</u>		417, 418, 422, 433, 434, 437,
<code>counter</code> (option)	<u>24</u>		441, 447–449, 466, 467, 473–
counters:			475, 479, 488, 490, 491, 494,
<code>glossaryentry</code>	<u>15</u>		496, 498, 500–502, 505, 506,
<code>glossarysubentry</code>	<u>16</u>		510, 514–517, 520–525, 535,
<code>counterwithin</code> (option)	<u>16</u>		537–539, 546–549, 555–559,
<code>\cs</code>	28, 47, 48, 58, 210,		564, 566–568, 571–573, 578,
	211, 229, 231–236, 242, 244,		580–582, 584, 585, 587–589,
	246, 250–252, 263, 264, 300,		595, 598, 599, 601, 603, 629–
	301, 328, 329, 350–354, 356–		632, 634–637, 640, 647, 648,
	358, 360–372, 374–376, 378–		665–668, 679, 684, 688, 689,
	396, 399, 402, 403, 405, 406,		716, 717, 723–725, 729, 738,
	408, 409, 1292, 1304, 1305,		740, 741, 744, 746, 748, 750,
	1408–1413, 1417, 1420, 1422, 1423		751, 753, 755, 756, 760, 761,
<code>\csdef</code> ...	26, 96–98, 108, 109, 116,		764–768, 771–776, 786, 788–
	117, 236, 237, 265, 277, 478,		790, 798–800, 806–811, 815,
	514–516, 522, 523, 526, 527,		816, 818, 819, 823–825, 829,
	589, 601, 606, 607, 672, 728,		832–834, 836–841, 847, 850–
	765, 766, 773, 774, 777, 840,		853, 855, 868, 882–885, 888–
	841, 853, 858, 926, 982, 1021,		890, 893, 900–902, 918–921,
	1022, 1029, 1030, 1034, 1096,		932, 937, 941, 942, 969, 970,
	1097, 1108, 1113, 1137, 1175–		977–979, 983, 994–997, 1000,
	1177, 1184, 1189, 1252, 1263,		1002, 1004, 1006–1008, 1011,
	1269, 1291, 1292, 1332–1334,		1012, 1016, 1017, 1020–1024,
	1341, 1346, 1413, 1414, 1429, 1434		1027–1031, 1033, 1042, 1043,
			1052–1054, 1060–1065, 1070,

1071, 1073, 1074, 1078–1080, 1085, 1087–1089, 1091, 1092, 1094, 1095, 1097, 1107, 1109, 1123–1125, 1132–1134, 1138, 1149–1152, 1154, 1155, 1157– 1159, 1161–1163, 1166, 1167, 1171, 1175–1179, 1181–1186, 1188, 1197, 1198, 1207–1209, 1216–1220, 1226, 1228–1230, 1233–1236, 1240, 1243–1245, 1247, 1248, 1250–1252, 1263, 1265, 1278–1280, 1286, 1288, 1293, 1304, 1307–1309, 1311– 1316, 1318–1320, 1323, 1324, 1328, 1332–1336, 1338–1343, 1345, 1354, 1355, 1364–1366, 1373–1377, 1383, 1385–1387, 1390–1393, 1397, 1400–1402, 1404, 1405, 1407, 1408, 1414, 1428, 1430, 1444–1446, 1453, 1454	977, 1104, 1132, 1260, 1286, 1425
\csshow 297, 631, 884, 1124, 1280, 1445	\CurrentOption 46, 421, 428, 488, 631, 683, 738, 884, 936, 994, 1148, 1149, 1304
\csuse 54, 57, 79, 87, 89, 90, 97, 98, 174, 200–202, 237, 246, 248, 250, 254, 277, 422, 491, 493, 505, 510, 511, 515, 516, 535, 536, 574, 575, 589, 591– 593, 595, 596, 601, 607, 673– 682, 684, 741, 743, 755, 760, 761, 766, 786, 826, 827, 841, 843–845, 847, 848, 853, 859, 927–931, 933–937, 997, 998, 1011, 1014, 1016, 1017, 1022, 1065, 1081, 1082, 1097, 1098, 1100–1103, 1113, 1152, 1153, 1166, 1169, 1171, 1172, 1176, 1177, 1220, 1236–1238, 1252, 1254–1256, 1258, 1259, 1269, 1309, 1310, 1322, 1326, 1328, 1329, 1333, 1334, 1377, 1393– 1395, 1414, 1420–1422, 1424, 1434	\CurrentTrackedLanguage 53, 420, 421, 491, 666, 683, 741, 920, 936, 997, 1151, 1308
\csxdef 107, 118, 521, 527, 772, 778, 868, 1028, 1035, 1183, 1190, 1340, 1347	\CurrentTrackedTag 53, 420, 491, 666, 683, 741, 919, 936, 997, 1151, 1308
\currentglossary 10, 15, 16, 57, 226, 300–302, 350, 473, 493, 587, 722, 723, 743, 839, 977, 998, 1095, 1125, 1131, 1132, 1153, 1250, 1286, 1310, 1407	\CustomAcronymFields 292, 628, 881, 882
\currentglssubentry 16, 256, 473, 597, 723, 849,	\CustomNewAcronymDef 292, 628, 629, 881, 882
	D
	\d 29, 479, 729, 983, 1138, 1293
	\datatool 28, 229, 232–234, 1292, 1293, 1408, 1411
	\datatoolasciistart 231, 1410
	\datatoolctrlboundary 231, 233, 234, 1410, 1411
	\day 189, 193, 566, 569, 668, 671, 818, 821, 922, 924, 1072, 1075, 1228, 1231, 1385, 1388
	debug (option) 7
	\DeclareAcronymList 22, 23, 24, 275, 276, 475– 477, 605, 606, 616, 618, 619, 621, 623, 625, 627, 629, 725– 727, 857, 858, 868, 870, 872, 874, 876, 878, 880, 882, 980, 981, 1112, 1113, 1134–1136, 1267, 1268, 1289, 1290, 1433, 1434
	\DeclareCurrentRelease 4, 299, 303, 306, 310, 316, 321, 327, 334, 342, 348, 419–421, 428, 467, 468, 718, 971, 1280, 1446, 1449
	\DeclareListParser 204, 577, 828, 1083, 1239, 1396
	\DeclareOption 12, 421, 428, 471, 631, 683, 721, 884, 936, 975, 1130, 1284
	\DeclareOptionX 12, 471, 721, 975, 1130, 1284
	\DeclareRelease 4, 299, 303, 306, 310, 316, 321, 327, 334, 342, 348, 419–421, 428, 467, 468, 718, 971, 1280, 1446, 1449

<code>\DeclareRobustCommand</code>	54,	1290, 1292–1294, 1298–1303,
	491, 624, 742, 876, 997, 1152, 1309	1306, 1307, 1309, 1310, 1313,
<code>\def</code> 12, 13, 15–20, 22, 23, 27–29, 36,		1315–1318, 1320, 1323–1327,
37, 39, 41, 42, 44, 49–51, 54,		1329, 1335, 1337–1341, 1343,
55, 62, 65–70, 75, 80–88, 91,		1344, 1346–1349, 1352, 1354–
100, 102, 104–109, 112, 113,		1362, 1364–1377, 1382, 1383,
117–122, 128, 132–142, 145–		1385, 1388–1395, 1397, 1398,
174, 184–186, 189, 193, 196–		1400, 1402–1405, 1407, 1413,
198, 200–203, 205–208, 211,		1416, 1420–1425, 1427–1430,
215, 216, 218–222, 225, 236,		1432, 1433, 1448, 1449, 1453, 1454
241, 247, 249, 250, 253–255,		
262, 264–269, 272–274, 302–		<code>\def@glsexdycheckbackslash</code>
305, 333, 334, 416–419, 421,	 142, 544,
424–427, 430–433, 447–454,		545, 796, 797, 1050, 1205, 1362
471–476, 479, 480, 484, 486–		<code>\DefaultNewAcronymDef</code> 289,
493, 495, 497–503, 505–509,		465, 615, 616, 661, 867, 869, 914
511, 517–524, 527–529, 535–		<code>\defglssdisplay</code> 127, 535, 786
559, 564, 566, 569–576, 578–		<code>\defglssdisplayfirst</code> . . . 127, 535, 786
580, 582–585, 587, 588, 592–		<code>\defglssentryfmt</code>
596, 600–605, 616–621, 623–		. . . 79, 81, 83, 276, 289, 505,
625, 627, 629, 632–634, 647–		506, 535, 606, 615, 616, 618,
652, 661–665, 668, 671, 673,		620, 622, 624, 626, 628, 755–
678, 679, 683, 685–690, 704,		757, 786, 858, 867, 869, 871,
705, 716, 717, 721–726, 728–		872, 874, 877, 879, 881, 1011,
730, 734–737, 739–743, 745,		1012, 1113, 1122, 1165–1167,
747–753, 755–759, 761, 767,		1269, 1278, 1322–1324, 1434, 1443
769–775, 777–780, 786–811,		<code>\define@boolkey</code> . . . 10, 13, 15, 16,
815, 816, 818, 820, 822–827,		21, 30, 31, 34–36, 42, 128, 129,
830–832, 834–840, 844–848,		254, 471–473, 475, 480, 481,
852–857, 868, 870–878, 880,		483, 484, 486, 536, 596, 720–
882, 885–888, 900–905, 915–		723, 725, 730, 733, 734, 736,
918, 921, 922, 924, 926, 932,		787, 848, 975–977, 980, 984,
936–940, 942, 943, 957, 958,		985, 987, 988, 991, 992, 1040,
969–971, 975–981, 983, 984,		1103, 1129–1132, 1134, 1139,
988, 990–995, 997–999, 1001,		1142, 1143, 1146, 1195, 1259,
1003–1008, 1011–1015, 1017,		1284–1286, 1289, 1294, 1295,
1024–1032, 1034–1037, 1040,		1297, 1298, 1301, 1302, 1352, 1424
1042–1065, 1070–1072, 1075,		<code>\define@choicekey</code> 7,
1077–1087, 1089–1094, 1096,		9–11, 17, 30, 31, 33, 36, 40,
1099, 1100, 1102–1104, 1106–		43, 45, 86, 253, 254, 469–471,
1112, 1125, 1130–1135, 1137–		473, 481, 482, 484, 485, 487,
1139, 1143, 1144, 1146, 1147,		508, 595, 596, 718, 720, 721,
1149, 1150, 1152, 1153, 1156,		723, 731–733, 736, 737, 758,
1158–1161, 1163, 1166–1170,		847, 848, 972, 974, 975, 977,
1172, 1178, 1180–1184, 1186,		985, 986, 988, 990, 992, 993,
1187, 1189–1192, 1195, 1197–		1014, 1103, 1127, 1129, 1132,
1205, 1207–1220, 1225, 1226,		1140, 1141, 1143, 1145, 1147,
1228, 1231–1238, 1240, 1241,		1148, 1169, 1258, 1259, 1281,
1243, 1245–1248, 1250, 1251,		1283, 1284, 1286, 1295, 1296,
1255–1259, 1262–1267, 1284–		1298, 1300, 1302, 1304, 1326, 1424
		<code>\define@key</code> 12, 16, 24,

32, 34, 37, 83–88, 96, 97, 128,
 185, 253, 255, 421, 430–432,
 471, 473, 477, 481, 484, 507–
 509, 514, 536, 564, 595, 596,
 632, 633, 683, 721, 723, 727,
 731, 734, 757–759, 765, 787,
 816, 847, 848, 886, 887, 936,
 975, 977, 981, 985, 987, 988,
 1013–1015, 1021, 1040, 1071,
 1102–1104, 1130, 1131, 1136,
 1140, 1142, 1143, 1167–1170,
 1175, 1176, 1195, 1226, 1258,
 1259, 1284, 1286, 1290, 1291,
 1295, 1297–1299, 1324–1327,
 1332, 1333, 1352, 1383, 1423, 1425
 \DefineAcronymSynonyms
 49, 287, 289,
 489, 615, 739, 867, 994, 1121,
 1122, 1149, 1277, 1306, 1442, 1443
 \delimN 57, 192, 204, 249,
 267, 493, 568, 577, 594, 602,
 670, 743, 820, 828, 846, 854,
 924, 998, 1074, 1084, 1102,
 1108, 1153, 1230, 1239, 1257,
 1264, 1310, 1387, 1396, 1422, 1429
 \delimR 57, 192, 266,
 267, 493, 568, 602, 670, 743,
 820, 854, 924, 998, 1074, 1108,
 1153, 1230, 1264, 1310, 1387, 1429
 description (key) 83
 description (option) 35
 descriptionaccess (key) 430
 \DescriptionDUANewAcronymDef . .
 290, 618, 620, 871, 872
 \DescriptionFootnoteNewAcronymDef
 290,
 465, 617, 618, 662, 869, 870, 915
 \descriptionname 51, 54,
 311–315, 317, 318, 323, 324,
 326, 336, 338–341, 343–348,
 490, 491, 693–696, 699–701,
 706–708, 710–712, 740, 741,
 946–950, 952–955, 959–966,
 995, 997, 1150, 1152, 1307, 1309
 \DescriptionNewAcronymDef . 290,
 465, 620, 621, 662, 873, 874, 916
 descriptionplural (key) 84
 descriptionpluralaccess (key) . . 431
 \DH 29, 480, 730, 984, 1139, 1294
 \dh 29, 480, 730, 984, 1139, 1294
 \dim 228, 352, 356–
 360, 364–377, 403–406, 408, 410
 \dimen@ . . 278, 279, 320, 415, 416,
 608, 697, 715, 716, 860, 951,
 969, 1115, 1270, 1436, 1452, 1453
 \directlua 47, 994, 1149, 1304
 \disable@keys
 . . . 49, 488, 738, 994, 1149, 1306
 disablemakegloss (option) 37
 \do 34, 46, 50, 52, 62, 64, 71, 94, 95,
 135, 184, 189–191, 202, 204,
 205, 214, 222, 236, 237, 276,
 483, 488, 489, 495, 496, 501,
 512, 513, 539, 564, 566, 567,
 576, 577, 582, 585, 589, 606,
 616, 618, 620, 621, 623, 626,
 627, 629, 668, 688, 689, 733,
 738, 740, 746, 751, 763, 764,
 790, 815, 818, 819, 828, 829,
 834, 837, 840, 841, 858, 868,
 871, 872, 874, 876, 879, 880,
 882, 922, 941, 942, 987, 994–
 996, 1001, 1002, 1007, 1019,
 1020, 1044, 1070, 1072–1074,
 1082–1084, 1089, 1093, 1096,
 1113, 1125, 1142, 1148, 1150,
 1151, 1156, 1157, 1161, 1162,
 1174, 1175, 1199, 1226, 1228,
 1229, 1238, 1239, 1244, 1248,
 1252, 1269, 1297, 1304, 1307,
 1313, 1314, 1318, 1319, 1331,
 1332, 1356, 1383, 1385, 1386,
 1395, 1396, 1401, 1405, 1413, 1434
 \do@glo@storeentry
 . . . 17–20, 108, 473–475, 522,
 723–725, 773, 977–979, 1029,
 1132–1134, 1184, 1287, 1288, 1341
 \do@gl@link@checkfirsthyper . .
 130,
 132, 145–151, 166–174, 447–
 449, 537, 546–549, 555–559,
 647, 648, 787, 788, 798–801,
 806–810, 900, 901, 1041, 1042,
 1051–1054, 1060–1065, 1196,
 1197, 1207–1210, 1216–1220,
 1353, 1354, 1364–1367, 1373–1377
 \do@gl@xdycheckbackslash
 . 135, 539, 790, 1044, 1199, 1356
 \do@gl@disablehyperinlist
 131, 132, 537,

788, 1042, 1196, 1197, 1353, 1354	476, 489, 491, 495–502, 505–
\do@glshaschildren 75, 502,	507, 510–513, 516–519, 523,
503, 752, 753, 1008, 1163, 1320	535, 537–545, 563–565, 569–
\do@write@glslabelsname	572, 574, 575, 577, 580, 583,
39, 39, 735, 990, 1144, 1145, 1300	584, 586, 588–590, 593, 597,
\doifglossarynoexistsordo	600, 602, 603, 615, 617, 618,
.. 74, 80, 502, 505, 752, 755,	620, 622, 624, 626, 661–664,
1008, 1011, 1163, 1166, 1320, 1323	667–669, 671, 688, 740, 741,
\dtl@ifsingle 600, 852	745–750, 755, 756, 760, 761,
\dtl@insertinto	763, 764, 786, 789–797, 816,
. 238, 590, 842, 1097, 1253, 1414	817, 820, 822–824, 826–828,
\dtl@sortresult .. 238, 239, 590,	832, 835, 836, 840, 842, 845,
842, 1097, 1098, 1253, 1414, 1415	852, 854, 855, 867, 869, 871,
\DTLassignlettergroup . . . 235, 1412	873, 875, 877, 879, 914–917,
\dtlcompare	920–922, 924, 995, 996, 1001–
. 239, 590, 842, 1098, 1253, 1415	1006, 1011, 1016, 1017, 1019,
\dtlcompare	1020, 1042–1050, 1071, 1075,
. 239, 590, 842, 1098, 1253, 1415	1077–1083, 1087, 1095, 1097,
\DTLifinlist	1098, 1101, 1107, 1109, 1150,
. 83, 131, 507, 537, 757, 788,	1151, 1155–1160, 1166, 1171,
1013, 1042, 1167, 1196, 1324, 1353	1173, 1175, 1197–1205, 1227,
\DTLifint	1231–1239, 1243, 1251, 1253,
. 262, 600, 852, 1107, 1262, 1427	1256, 1262, 1264, 1265, 1307,
\dtlletterindexcompare	1308, 1312–1317, 1323, 1328,
. 238, 590, 842, 1097, 1253, 1414	1330, 1332, 1354–1362, 1384,
\dtlnonlettergroup 234, 1411	1388–1392, 1396, 1400, 1408,
\dtlnumbergroup 234, 1411	1414, 1415, 1421, 1427, 1430
\DTLsortedletter 234, 235, 1412	\egroup 28,
\DTLsortlettercasehandler 233	103, 185, 220, 237, 267, 479,
\DTLsortletterhandler . . . 232, 1411	518, 564, 587, 589, 729, 769,
\DTLsortwordcasehandler . 233, 1411	816, 839, 841, 983, 1025, 1070,
\DTLsortwordhandler	1092, 1095, 1096, 1109, 1138,
. 232, 234, 1410, 1411	1180, 1226, 1247, 1252, 1264,
\DTLsubstituteall	1293, 1337, 1383, 1404, 1414, 1430
. 135, 540, 790, 1044, 1199, 1356	\else .. 8, 15, 16, 21–23, 25, 27, 30,
\dtlwordindexcompare	31, 39, 41, 42, 46, 49, 50, 55,
. 238, 590, 842, 1097, 1253, 1414	56, 59, 61–71, 79, 86, 89, 105,
dua (acrstyle) 282, 459	106, 109–112, 117, 118, 129,
dua (option) 35	131–142, 177, 188, 189, 192–
dua-desc (acrstyle) 284, 461	196, 198–200, 202–204, 212–
\DUANewAcronymDef	216, 218–222, 226, 248, 254,
. 291, 626, 627, 879, 880	256, 257, 264, 266, 267, 279,
	291, 293, 307, 311, 312, 314,
	317, 318, 320, 322, 324, 325,
	336, 337, 339, 343, 345, 347,
	412, 413, 415, 417, 418, 438,
	469, 470, 472, 473, 475–478,
	480, 481, 485–489, 492–501,
	508, 510, 520, 521, 523, 524,
	527, 537–549, 560, 565, 566,
E	
\eappto 506, 507, 523, 582	
\edef 50, 53,	
62–67, 69, 81, 89, 90, 93, 95,	
133–142, 187, 188, 193, 195–	
198, 204, 212, 227, 228, 238,	
239, 248, 262, 267, 269, 475,	

568–575, 580–585, 587, 593,	1256, 1387, 1417, 1420, 1446–1449
596–598, 600, 602, 608, 618,	<code>\end@doifinlist</code>
619, 621, 623, 624, 626–629,	.. 62, 495, 745, 1001, 1156, 1313
637, 668–673, 677–679, 688,	<code>\end@getprefix</code>
691, 693, 694, 696–700, 705–	. 221, 584, 836, 1092, 1247, 1404
707, 709–711, 713–717, 719,	<code>\end@glis@islistofacronyms</code> .. 23,
722–728, 730, 731, 735–740,	476, 477, 726, 980, 981, 1135, 1290
742–751, 758, 760, 771, 772,	<code>\EndAccSupp</code> 429, 632, 885
774, 775, 778, 788–800, 811,	<code>\endcsname</code> 17, 20, 29, 46, 51, 53, 54,
812, 817, 818, 820, 822–827,	60, 63, 66, 69, 71, 75, 81, 82,
832–837, 839, 845, 848–850,	89, 90, 96–98, 100, 101, 105–
852, 854, 860, 870–872, 874,	112, 114, 115, 132–134, 146–
876–879, 881, 882, 891, 921,	150, 167–174, 184, 186, 190,
922, 924–926, 930–932, 941,	191, 197–200, 206, 209, 212,
944, 946–954, 959–961, 963–	213, 215, 220, 221, 225–227,
965, 967–971, 973, 976–982,	238, 264, 265, 269, 293–299,
984, 985, 990, 991, 994, 995,	417, 418, 422, 433, 434, 437,
998–1007, 1011, 1014, 1016,	441, 447–449, 466, 467, 473–
1027, 1028, 1030, 1031, 1035,	475, 479, 488, 490, 491, 494,
1040–1050, 1066, 1072, 1073,	496, 498, 500–502, 505, 506,
1075–1077, 1079–1083, 1088–	510, 514–517, 520–525, 535,
1093, 1095, 1100, 1104, 1105,	537–539, 546–549, 555–559,
1107–1109, 1115, 1122, 1125,	564, 566–568, 571–573, 578,
1128, 1131–1137, 1139, 1140,	580–582, 584, 585, 587–589,
1145, 1146, 1148–1150, 1152–	595, 598, 599, 601, 603, 629–
1162, 1165, 1169, 1171, 1181,	632, 634–637, 640, 647, 648,
1182, 1184–1186, 1189, 1190,	665–668, 679, 684, 688, 689,
1195–1205, 1222, 1227, 1228,	716, 717, 723–725, 729, 738,
1230–1238, 1243–1248, 1250,	740, 741, 744, 746, 748, 750,
1256, 1259–1261, 1263–1265,	751, 753, 755, 756, 760, 761,
1270, 1278, 1282, 1283, 1285,	764–768, 771–776, 786, 788–
1286, 1289–1292, 1294, 1295,	790, 798–800, 806–811, 815,
1300–1302, 1304, 1306, 1307,	816, 818, 819, 823–825, 829,
1309–1319, 1322, 1326, 1328,	832–834, 836–841, 847, 850–
1338, 1339, 1341–1343, 1346,	853, 855, 868, 882–885, 888–
1347, 1352–1362, 1379, 1384,	890, 893, 900–902, 918–921,
1385, 1387–1395, 1400–1405,	932, 937, 941, 942, 969, 970,
1407, 1421, 1424–1426, 1428–	977–979, 983, 994–997, 1000,
1430, 1436, 1443, 1444, 1450–1454	1002, 1004, 1006–1008, 1011,
<code>\emph</code> . 222, 268, 585, 603, 837, 855,	1012, 1016, 1017, 1020–1024,
1093, 1109, 1248, 1265, 1405, 1430	1027–1031, 1033, 1042, 1043,
<code>\empty</code> 267, 428, 602,	1052–1054, 1060–1065, 1070,
631, 854, 885, 1109, 1264, 1430	1071, 1073, 1074, 1078–1080,
<code>\encodingdefault</code>	1085, 1087–1089, 1091, 1092,
... 29, 479, 729, 983, 1138, 1293	1094, 1095, 1097, 1107, 1109,
<code>\end</code> 192, 243, 247, 307, 310–	1123–1125, 1132–1134, 1138,
313, 315, 316, 318–326, 329–	1149–1152, 1154, 1155, 1157–
348, 568, 593, 670, 691–695,	1159, 1161–1163, 1166, 1167,
697–712, 819, 844, 923, 944,	1171, 1175–1179, 1181–1186,
946–966, 1074, 1100, 1230,	1188, 1197, 1198, 1207–1209,

1216–1220, 1226, 1228–1230,	408, 409, 1287, 1288, 1292,
1233–1236, 1240, 1243–1245,	1293, 1305, 1409–1412, 1418–1420
1247, 1248, 1250–1252, 1263,	<code>\expandafter</code> 9, 20, 27–29,
1265, 1278–1280, 1286, 1288,	46, 50, 53, 63, 66, 68–71, 75,
1293, 1304, 1307–1309, 1311–	81–83, 89, 90, 94–98, 100, 101,
1316, 1318–1320, 1323, 1324,	105–115, 127, 131, 133–142,
1328, 1332–1336, 1338–1343,	175–177, 180, 185, 187, 188,
1345, 1354, 1355, 1364–1366,	191, 196, 197, 199, 212–214,
1373–1377, 1383, 1385–1387,	216, 218–220, 222, 226, 227,
1390–1393, 1397, 1400–1402,	236, 238, 247, 258, 265, 267,
1404, 1405, 1407, 1408, 1414,	269, 293–299, 404, 410, 422,
1428, 1430, 1444–1446, 1453, 1454	433, 434, 437, 438, 465–467,
<code>\endfoot</code> 311–313, 315,	470, 474, 475, 479, 488–491,
317, 318, 322–324, 326, 693–	496, 498–502, 505–507, 510–
696, 699–701, 946–950, 952–955	517, 520–525, 535, 537–545,
<code>\endgroup</code> 7, 9, 39, 214,	560, 564, 565, 567, 568, 570–
219, 227, 267, 468, 470, 581,	573, 580–585, 587, 589, 593,
583, 602, 718, 720, 735, 833,	594, 598, 599, 601–603, 617,
835, 854, 972, 974, 990, 1088,	622, 629–632, 634–638, 661,
1091, 1109, 1127, 1128, 1145,	665–668, 670, 671, 684, 688,
1244, 1246, 1251, 1264, 1281,	689, 716, 719, 723–725, 728,
1283, 1300, 1401, 1403, 1408, 1430	729, 738, 740, 741, 746, 748,
<code>\endhead</code> 311–315,	750, 751, 753, 755–757, 760,
317, 318, 322–324, 326, 693–	761, 763–768, 771–776, 786,
696, 699–701, 946–950, 952–955	788–796, 811–813, 816, 817,
<code>\endtheglossary</code>	819, 822–825, 832–837, 839,
. . 6, 7, 468, 718, 972, 1127, 1281	841, 845, 846, 850, 851, 853–
<code>entrycounter</code> (option) <u>16</u>	855, 868, 869, 875, 882–885,
<code>\entryname</code> <u>51</u> , 54,	888–891, 914, 918–921, 923,
311–315, 317, 318, 323, 324,	924, 937, 941, 942, 969, 974,
326, 336, 338–341, 343–348,	978, 979, 983, 994–996, 1002,
490, 491, 693–696, 699–701,	1004–1008, 1011–1013, 1016,
706–708, 710–712, 740, 741,	1017, 1019–1024, 1027–1033,
946–950, 952–955, 959–966,	1040, 1042–1050, 1065, 1066,
995, 997, 1150, 1152, 1307, 1309	1068, 1070, 1071, 1074, 1077–
environments:	1080, 1087–1091, 1093, 1095,
<code>theglossary</code> <u>257</u>	1097, 1100, 1101, 1105, 1107,
<code>\epreto</code> 112, 1031, 1186, 1343	1109, 1123–1126, 1128, 1132–
<code>\equal</code> 32, 50, 60, 133, 204, 481, 489,	1134, 1137, 1138, 1149–1151,
494, 538, 576, 600, 688, 731,	1156–1163, 1166, 1167, 1171,
739, 744, 789, 828, 852, 941,	1172, 1174–1179, 1181–1188,
985, 995, 1000, 1042, 1083,	1194–1205, 1221–1223, 1226,
1125, 1140, 1149, 1155, 1197,	1227, 1229, 1230, 1233–1235,
1239, 1295, 1306, 1312, 1354, 1396	1243–1247, 1249–1253, 1256,
<code>esclocations</code> (option) <u>13</u>	1257, 1261, 1263–1265, 1278–
<code>\everypar</code> 403, 406, 410	1280, 1283, 1288, 1292, 1293,
<code>\exp</code> 18, 19, 28, 210, 229–235,	1304, 1307, 1308, 1313–1320,
244–246, 251, 252, 303, 328,	1323, 1324, 1328, 1329, 1331–
329, 349, 352, 355–358, 363,	1336, 1338–1345, 1351–1362,
365–369, 375–377, 402–404,	1378–1380, 1383, 1384, 1386,

1387, 1390–1392, 1400–1405,	520–524, 527, 537–549, 560,
1407, 1408, 1413, 1414, 1420,	564–566, 568, 570–574, 576,
1426, 1428, 1430, 1444–1446, 1453	577, 581–588, 593, 596–598,
<code>\expandonce</code> 89–91,	600–603, 608, 615, 616, 618,
101, 135, 136, 197, 198, 215,	619, 621–629, 631, 632, 638,
238, 239, 275, 510, 511, 539,	667–673, 677–679, 688, 691,
540, 571–573, 582, 590, 598,	693, 694, 696–700, 705–707,
599, 605, 615, 617, 619, 621,	709–711, 713–728, 730–732,
622, 625, 626, 632, 760, 761,	735–740, 742–751, 756, 758,
768, 790, 791, 823, 824, 834,	760, 768, 770–775, 778, 788–
842, 850, 851, 857, 867–871,	801, 812, 816–818, 820, 822–
873, 875, 878, 879, 885, 1016,	829, 832–840, 845, 848–850,
1017, 1024, 1044, 1045, 1078,	852–854, 860, 867, 868, 870–
1079, 1089, 1097, 1098, 1112,	872, 874, 876–882, 884, 885,
1171, 1172, 1179, 1199, 1200,	891, 920–922, 924–926, 930–
1233–1235, 1245, 1253, 1268,	932, 941, 944, 946–954, 959–
1328, 1329, 1336, 1356, 1357,	961, 963–965, 967–982, 984,
1390–1392, 1402, 1414, 1415, 1433	985, 987–995, 997–1007, 1011,
<code>\ExplSyntaxOff</code> 6, 12, 20,	1012, 1014, 1016, 1024, 1026–
28, 31, 49, 59, 72, 76, 112, 143,	1031, 1034, 1035, 1040–1050,
211, 236, 246, 252, 262–264,	1066, 1071–1073, 1075–1077,
303, 329, 410, 416, 972, 1106,	1079–1084, 1088–1096, 1101,
1107, 1126, 1186, 1206, 1262,	1103–1105, 1107–1109, 1115,
1263, 1281, 1288, 1293, 1306,	1122, 1123, 1125, 1127–1137,
1343, 1363, 1413, 1420, 1423, 1428	1139, 1140, 1142, 1143, 1145–
<code>\ExplSyntaxOn</code> 4, 12, 17,	1150, 1152–1162, 1165, 1167,
27, 30, 47, 58, 72, 76, 112, 143,	1169, 1171, 1179, 1181–1183,
210, 228, 242, 250, 262, 263,	1185, 1186, 1189, 1190, 1195–
300, 327, 348, 416, 971, 1106,	1205, 1222, 1226–1228, 1230,
1107, 1126, 1186, 1206, 1262,	1232, 1233, 1235, 1237–1239,
1280, 1287, 1292, 1304, 1343,	1243–1251, 1256, 1259–1261,
1363, 1408, 1417, 1422, 1427, 1428	1263–1265, 1270, 1277, 1278,
	1281–1287, 1289–1292, 1294,
F	1295, 1297–1304, 1306, 1307,
<code>\fi</code> 7, 8, 10,	1309–1319, 1322, 1324, 1326,
11, 15–17, 21–23, 25, 27, 30,	1328, 1336, 1338–1340, 1342,
31, 33, 35, 37, 39–42, 44–46,	1343, 1346, 1347, 1352–1362,
49, 50, 54–56, 59, 61–71, 79,	1379, 1383–1385, 1387, 1389,
82, 86, 89, 101, 105–112, 117,	1390, 1392, 1394–1396, 1400–
118, 129, 131–142, 177, 186,	1408, 1421, 1424–1426, 1428–
188, 189, 192, 194–196, 198,	1430, 1436, 1443, 1444, 1450–1454
199, 201–205, 212–216, 218–	<code>\findrootlanguage</code> 69, 500, 750
223, 225–228, 248, 254, 256,	<code>first</code> (key) <u>84</u>
257, 264, 266–268, 279, 289,	<code>firstaccess</code> (key) <u>430</u>
291, 293, 307, 311, 312, 314,	<code>\firstacronymfont</code>
317, 318, 320–322, 324, 325, 127, <u>274</u> , 277–279, 285,
336, 337, 339, 343, 345, 347,	446, 447, 456, 457, 462, 535,
404, 410, 412, 413, 415–418,	605, 607, 608, 613, 617, 620,
429, 438, 468–481, 483, 485–	622, 624, 647, 653, 654, 658,
489, 491–501, 506, 508, 510,	785, 786, 857, 859, 860, 865,

869, 873, 875, 877, 900, 906,
 907, 911, 1040, 1111, 1114,
 1115, 1119, 1120, 1194, 1195,
 1267, 1269–1271, 1275, 1351,
 1352, 1432, 1435, 1436, 1440, 1441
 firstplural (key) 84
 firstpluralaccess (key) 430
 \fmtversion
 ... 29, 480, 730, 984, 1139, 1294
 \footnote
 285, 289, 290, 462, 613, 616,
 658, 865, 869, 911, 1119, 1120,
 1122, 1275, 1278, 1440, 1441, 1443
 footnote (acrstyle) 284, 462
 footnote (option) 35
 footnote-desc (acrstyle) ... 286, 464
 footnote-sc (acrstyle) 286, 463
 footnote-sc-desc (acrstyle) . 287, 464
 footnote-sm (acrstyle) 286, 464
 footnote-sm-desc (acrstyle) . 287, 464
 \FootnoteNewAcronymDef ... 291,
 465, 622, 623, 663, 875, 876, 917
 \footnotesize
 8, 470, 719, 973, 1128, 1283
 \forallacronyms
 .. 71, 501, 751, 1007, 1162, 1318
 \forallallglossaries
 70, 71, 211, 225, 416,
 501, 580, 586, 715, 751, 832,
 838, 969, 1007, 1087, 1094,
 1161, 1162, 1243, 1249, 1250,
 1318, 1319, 1400, 1406, 1407, 1452
 \forallallglsentries
 39, 71, 115, 116, 119,
 186, 187, 485, 501, 525, 526,
 528, 565, 735, 751, 776–778,
 816, 990, 1007, 1033–1035,
 1071, 1144, 1162, 1188, 1190,
 1227, 1300, 1319, 1345, 1347, 1384
 \ForEachTrackedDialect
 53, 420, 491, 666,
 683, 741, 919, 936, 997, 1151, 1308
 \forallorglsentries
 ... 71, 71, 75, 109, 241, 416,
 501, 502, 523, 592, 715, 751,
 753, 774, 844, 969, 1007, 1008,
 1030, 1099, 1162, 1163, 1184,
 1255, 1319, 1320, 1341, 1416, 1452
 \foralllistcsloop 229,
 236, 243, 247, 251, 589, 593,
 840, 844, 1096, 1100, 1252,
 1256, 1409, 1413, 1417, 1420, 1423
 \foralllistloop 208, 249,
 579, 594, 831, 846, 1086, 1102,
 1241, 1242, 1257, 1398, 1399, 1422

G
 \g 210, 211, 251, 252
 \gdef 39, 63, 80, 81, 100,
 105, 106, 213, 255, 302, 474,
 496, 505, 516, 520, 581, 596,
 597, 688, 724, 735, 746, 755,
 767, 771, 833, 848, 941, 978,
 990, 1002, 1011, 1023, 1027,
 1088, 1104, 1125, 1133, 1145,
 1157, 1166, 1178, 1181, 1182,
 1244, 1259, 1260, 1300, 1314,
 1323, 1335, 1338, 1339, 1401, 1425
 \Genacrformat
 126, 127, 275, 277–
 279, 285, 446, 446, 456, 457,
 462, 534, 535, 606–608, 613,
 646, 647, 653, 654, 658, 785,
 857–860, 865, 899, 900, 906,
 907, 911, 1039, 1040, 1112,
 1114, 1115, 1119, 1194, 1268–
 1271, 1275, 1351, 1433–1436, 1440
 \genacrformat
 126, 126, 127, 275,
 277–279, 285, 446, 446, 456,
 462, 534, 535, 605–608, 613,
 646, 653, 654, 658, 785, 786,
 857–860, 865, 899, 900, 906,
 907, 911, 1039, 1040, 1112,
 1114, 1115, 1119, 1194, 1268–
 1270, 1275, 1351, 1433–1436, 1440
 \GenericAcronymFields
 275, 276, 277–
 287, 456–462, 464, 465, 605–
 615, 653–655, 657, 658, 660,
 857–866, 906–911, 913, 914,
 1112–1119, 1121, 1268–1270,
 1272–1275, 1277, 1433–1440, 1442
 \Genplacrformat
 . 126, 127, 275, 277–279, 285,
 446, 447, 456, 457, 462, 534,
 535, 606–608, 613, 646, 647,
 653, 654, 658, 785, 786, 858–
 860, 865, 899, 900, 906, 907,
 911, 1039, 1040, 1112–1115,

1120, 1194, 1195, 1268–1271, 1275, 1351, 1352, 1433–1436, 1441	\glosortentrieswarning 25, <u>25</u> , 236, 477, 478, 588, 727, 840, 981, 982, 1096, 1136, 1251, 1291, 1413
\genplacrformat . . . 126, 127, <u>127</u> , 275, 277–279, 285, 445, 446, <u>447</u> , 456, 457, 462, 534, 535, 606–608, 613, 646, 647, 653, 654, 658, 785, 786, 858– 860, 865, 899, 900, 906, 907, 911, 1039, 1040, 1112–1115, 1120, 1194, 1195, 1268–1271, 1275, 1351, 1352, 1433–1436, 1441	\glossaries . . 58, 72, 76, 230, 231, 243, 263, 264, 300–302, 349– 354, 356–358, 361, 362, 367, 368, 370–372, 387, 388, 390, 392, 393, 399–404, 406–409, 416
\GetTitleStringDisableCommands 306, 944	\GlossariesWarning 7, 10, <u>24</u> , 25, 30–32, 38, 41, 43, 44, 52, 58, 61, 62, 74, 78, 79, 86, 89, 92–94, 119–121, 184, 199–202, 206–211, 214, 215, 221, 224, 225, 291, 428, 469, 471, 477, 478, 480, 481, 484–487, 493, 495, 502, 504, 508, 510, 528, 529, 535, 536, 563, 574, 575, 577–581, 584, 586, 587, 599, 601, 631, 667, 718–720, 727, 728, 730, 731, 734–737, 743, 745, 752, 755, 758, 760, 762, 763, 779, 786, 787, 815, 826, 827, 829–833, 836, 838, 839, 851, 853, 880, 885, 920, 972–974, 981, 982, 984, 985, 989, 990, 992, 996, 999, 1001, 1008, 1010, 1011, 1014, 1016, 1018, 1019, 1036, 1070, 1080–1082, 1084–1089, 1092, 1094, 1122, 1127–1129, 1136, 1137, 1139, 1140, 1143– 1145, 1147, 1151, 1153, 1155, 1162, 1165, 1169, 1170, 1173, 1174, 1191, 1225, 1235–1238, 1240–1242, 1244, 1245, 1248– 1250, 1278, 1282, 1284, 1291, 1294, 1295, 1299, 1301, 1302, 1308, 1310, 1312, 1319, 1322, 1326, 1327, 1330, 1331, 1348, 1382, 1392–1395, 1397–1399, 1401, 1402, 1404, 1406, 1407, 1443
\GetTitleStringSetup 306, 944	\GlossariesWarningNoLine 7, 8, 25, <u>25</u> , 47, 205, 206, 212, 228, 302, 469, 477, 478, 576–578, 580, 588, 688, 718, 719, 727, 829, 830, 832, 840, 941, 972, 973, 981, 982, 1084, 1085, 1087, 1088, 1096, 1125, 1127, 1128, 1136, 1240, 1243, 1251, 1282,
\glo@desc 673, 926	
\glo@do@compare . . 238, 239, 590, 842, 1097, 1098, 1253, 1414, 1415	
\glo@grabfirst 236, <u>247</u> , 593, 594, 845, 846, 1100, 1101, 1256, 1257, 1413, 1420	
\glo@label 75, 109, 502, 523, 753, 774, 1008, 1030, 1163, 1184, 1320, 1341	
\glo@list 109, 523, 774, 1030, 1184, 1341	
\glo@name 258, 598, 850, 1105, 1261, 1426	
\glo@parent 75, 502, 753, 1008, 1163, 1320	
\glo@type 109, 523, 774, 1030, 1184, 1341	
\glo@value 94, 512, 763, 1019, 1174, 1331	
\global . 20, 89, 92, 102, 108, 115, 213, 227, 248, 249, 255, 264, 320, 321, 474, 475, 510, 512, 518, 522, 525, 581, 587, 593, 594, 597, 698, 724, 725, 760, 762, 769, 773, 776, 833, 839, 845, 846, 849, 951, 978, 979, 1016, 1018, 1025, 1029, 1033, 1088, 1095, 1101, 1104, 1133, 1134, 1171, 1173, 1180, 1184, 1188, 1244, 1251, 1256, 1257, 1260, 1288, 1328, 1330, 1337, 1341, 1345, 1401, 1408, 1421, 1425	
\glo@linkprefix 132, <u>133</u> , 185, 258, 538, 564, 598, 789, 816, 850, 1042, 1071, 1105, 1197, 1226, 1261, 1354, 1383, 1426	

1291, 1304, 1396, 1397, 1400, 1408	567, 592, 670, 743, 819, 844,
<code>\glossary</code>	923, 999, 1074, 1100, 1153,
43, <u>43</u> , 44, 486, 487, 668, 736,	1229, 1255, 1310, 1386, 1417, 1420
737, 921, 992, 1147, 1302, 1303	<code>\glossarystyle</code> <u>265</u> , 601, 853
<code>\glossaryentry</code> 220, 221,	<code>glossarysubentry</code> (counter) <u>16</u>
584, 585, 668, 836, 837, 921,	<code>\glossarysubentryfield</code>
1091, 1092, 1247, 1248, 1404, 1405 <u>261</u> , 455, 599, 600,
<code>glossaryentry</code> (counter) <u>15</u>	653, 673–682, 851, 906, 926–935
<code>\glossaryentryfield</code> <u>260</u> , 455, 598,	<code>\glossarytitle</code> . . . 191, 225, 226,
599, 653, 673–678, 680–682,	243, 246, 253, 567, 587, 592,
850, 851, 906, 926–931, 933–935	595, 670, 819, 838, 839, 844,
<code>\glossaryentrynumbers</code>	847, 923, 1074, 1094, 1095,
. . 13, <u>13</u> , 192, 226, 227, 244,	1100, 1102, 1229, 1250, 1255,
246, 248, 249, 254, 255, 471,	1258, 1386, 1407, 1417, 1420, 1423
472, 568, 587, 594, 596, 597,	<code>\glossarytoctitle</code>
670, 721, 820, 839, 845, 846,	11, 21, 22, 44, 45, 51, 54, 59,
848, 849, 923, 975, 1074, 1094,	191, 225, 243, 246, 253, 254,
1095, 1101, 1104, 1130, 1230,	471, 475, 476, 487, 488, 490,
1250, 1251, 1257, 1259, 1260,	491, 493, 567, 587, 592, 595,
1284, 1285, 1387, 1407, 1408,	596, 670, 721, 725, 726, 738,
1418, 1419, 1421, 1422, 1424, 1425	740, 741, 743, 819, 838, 844,
<code>\glossaryheader</code> 192, 243,	847, 848, 923, 975, 979, 980,
247, <u>258</u> , 304, 307–315, 317,	993, 995, 997, 999, 1074, 1094,
318, 322–326, 328, 330–333,	1100, 1103, 1130, 1134, 1135,
335, 337, 338, 342, 344, 346,	1148, 1150, 1152, 1153, 1229,
402, 411–415, 417, 419, 568,	1250, 1255, 1258, 1259, 1284,
593, 598, 670, 689, 691–696,	1289, 1303, 1307, 1309, 1310,
698–707, 709–711, 713–717,	1386, 1407, 1417, 1420, 1423, 1424
819, 844, 850, 923, 942, 944–	<code>\glossentry</code> 109, 226, 246,
950, 952–961, 963–971, 1074,	249, <u>258</u> , 304, 307–312, 314,
1100, 1105, 1230, 1256, 1261,	322, 323, 325, 335, 337, 339,
1387, 1417, 1420, 1426, 1446–1454	342, 344, 346, 402, 411, 413,
<code>\glossarymark</code> 59, <u>59</u> , 493,	414, 417, 523, 587, 594, 599,
743, 744, 999, 1154, 1310, 1311	689, 691–694, 698–700, 705–
<code>\glossaryname</code>	707, 709–711, 713–716, 774,
21, <u>51</u> , 53, 54, 475, 490, 491,	839, 846, 851, 942, 944–948,
725, 740, 741, 979, 995–997,	952–954, 959–961, 963–968,
1134, 1150–1152, 1289, 1307–1309	970, 1030, 1095, 1101, 1105,
<code>\glossarypostamble</code>	1185, 1250, 1257, 1261, 1342,
. <u>58</u> , 192, 243, 247, 493,	1407, 1419, 1421, 1426, 1450–1453
568, 593, 670, 743, 819, 845,	<code>\GLOSSentrydesc</code> <u>259</u> , 386, 387
923, 999, 1074, 1100, 1153,	<code>\GlossEntryDesc</code> <u>259</u> , 386, 387
1230, 1256, 1310, 1387, 1417, 1420	<code>\Glossentrydesc</code> . . 143, <u>259</u> , 386,
<code>\glossarypreamble</code>	387, 455, 599, 652, 851, 905,
. <u>57</u> , 191, 243, 246, 493,	1106, 1206, 1261, 1262, 1363, 1427
567, 592, 670, 743, 819, 844,	<code>\glossentrydesc</code> 143, 259, <u>259</u> , 304,
923, 998, 1074, 1100, 1153,	305, 307–309, 311, 312, 314,
1229, 1255, 1310, 1386, 1417, 1420	322, 323, 325, 335, 337, 339,
<code>\glossarysection</code>	342–344, 346, 362, 386, 411–
. <u>58</u> , 191, 243, 246, 493,	415, 417, 418, 455, 599, 652,

690–694, 698–700, 705–707, 709–711, 713–717, 850, 905, 943–948, 952–954, 959–961, 963–965, 967–970, 1106, 1206, 1261, 1262, 1363, 1427, 1450–1454	\gls@accessibility
\GLOSSentryname 258, 383, 384 429, 429, 632, 885, 886
\GlossEntryName 259, 383, 384	\gls@accsupp@engine 429, 429, 632, 885
\Glossentryname	\gls@Alphpage 214,
143, 258, 382, 383, 598, 850, 1106, 1206, 1261, 1363, 1426, 1427	214, 219, 582, 583, 833–835, 1089, 1091, 1244–1246, 1401, 1403
\glossentryname 143, 258, 258, 261, 304, 306–312, 314, 322, 323, 325, 335, 337, 339, 342, 344, 346, 361, 382, 383, 411–414, 417, 418, 454, 455, 598, 652, 689–694, 698–700, 705–707, 709–711, 713–717, 850, 905, 943–948, 952–954, 959–961, 963–965, 967, 968, 970, 1105, 1106, 1206, 1261, 1262, 1363, 1426, 1427, 1450–1454	\gls@alphpage 214, 214, 219, 582, 583, 833–835, 1089, 1091, 1244–1246, 1401, 1403
\glossentrynameother 143, 1206, 1363	\gls@arabicpage 214, 215, 219, 582, 583, 833–835, 1089, 1090, 1244–1246, 1401, 1403
\GLOSSentrysymbol 260, 384, 385	\gls@assign@desc 102, 102, 108, 518, 522, 769, 773, 1025, 1029, 1180, 1183, 1337, 1340
\GlossEntrySymbol 260, 385	\gls@assign@desc@field 26
\Glossentrysymbol 143, 260, 384, 385, 455, 599, 652, 851, 906, 1106, 1206, 1262, 1363, 1427	\gls@assign@descplural 616, 623, 625, 627, 661, 664, 665, 868, 875, 876, 878, 880, 915, 917, 918
\glossentrysymbol 143, 259, 260, 304, 305, 314, 325, 339, 346, 362, 384, 385, 411–414, 417, 418, 455, 599, 652, 689, 690, 694, 700, 707, 711, 713– 717, 851, 906, 943, 948, 954, 961, 965, 967, 968, 970, 1106, 1206, 1262, 1363, 1427, 1450–1454	\gls@assign@descplural@field . . 26
\GLS . . . 147, 547, 798, 1052, 1208, 1365	\gls@assign@field
\Gls 120, 146, 146, 269, 288, 528, 546, 603, 615, 779, 798, 855, 867, 1036, 1052, 1109, 1122, 1191, 1207, 1265, 1277, 1348, 1364, 1430, 1443	91, 91, 96, 97, 102, 104, 106– 108, 422, 511, 514, 518, 519, 521, 522, 684, 761, 765, 769– 773, 937, 1017, 1021, 1025– 1029, 1172, 1175, 1176, 1180– 1183, 1329, 1332, 1333, 1337–1340
\gls . . . 43, 119, 145, 146, 206, 256, 269, 288, 487, 528, 546, 577, 597, 603, 615, 737, 779, 798, 829, 849, 855, 867, 992, 1036, 1051, 1052, 1084, 1105, 1109, 1122, 1147, 1191, 1207, 1240, 1260, 1265, 1277, 1302, 1348, 1364, 1397, 1426, 1430, 1431, 1443	\gls@assign@firstpl 616–619, 621, 623, 625–627, 661–665, 867, 868, 870–876, 878, 880, 914–918
\gls@@delimN . . . 267, 1108, 1264, 1429	\gls@assign@name@field 26
	\gls@assign@plural 616–619, 621, 623, 625–627, 661–665, 868, 870–876, 878, 880, 915–918
	\gls@assign@symbolplural
 616–619, 621, 625–627, 662, 663, 665, 868, 870–874, 878, 880, 915, 916, 918
	\gls@assign@type@field 26
	\gls@automake@makegloss . 40, 41, 42, 202, 203, 990, 991, 1083, 1145, 1146, 1238, 1300, 1301, 1395
	\gls@automake@nr
	40, 41, 42, 48, 202, 485, 576, 736, 827, 990, 991, 1082, 1145, 1146, 1238, 1300, 1301, 1305, 1395
	\gls@automake@option
 40, 40, 41, 42, 990, 991, 1145, 1146, 1300, 1301

<code>\gls@automake@val</code>	40–42, 485, 736, 990, 991, 1145, 1146, 1300, 1301	<code>\gls@dosubst</code>	135, 540, 790, 791, 1044, 1199, 1356
<code>\gls@begindocdefs</code>	93, 93, 512, 763, 1019, 1173, 1330	<code>\gls@dotoctitle</code>	225, 226, 253, 587, 595, 839, 847, 1094, 1095, 1102, 1103, 1250, 1258, 1407, 1423
<code>\gls@checkisacronymlist</code>	23, 131, 477, 537, 726, 788, 981, 1041, 1135, 1196, 1290, 1353	<code>\gls@end@sanitizesort</code>	27, 28, 479, 728, 729, 983, 1137, 1138, 1292, 1293
<code>\gls@checkseeallowed</code>	86, 86, 204, 206, 508, 511, 576, 577, 758, 828, 829, 1014, 1083, 1085, 1169, 1239, 1240, 1325, 1326, 1396, 1397	<code>\gls@endcheck</code>	90, 91, 511, 761, 1017, 1172, 1329
<code>\gls@checkseeallowed@preambleonly</code>	86, 508, 511, 758, 1014, 1169, 1326	<code>\gls@glossary</code>	43, 44, 212, 220, 221, 487, 581, 584, 585, 737, 833, 836, 837, 992, 993, 1088, 1091, 1092, 1147, 1243, 1247, 1248, 1302, 1303, 1400, 1404, 1405
<code>\gls@codepage</code>	37, 70, 200, 201, 228, 484, 500, 574, 575, 588, 734, 750, 751, 826, 827, 840, 988, 1006, 1080, 1081, 1096, 1143, 1161, 1236, 1237, 1251, 1298, 1299, 1318, 1393, 1394, 1408	<code>\gls@glsdefs@addlabel</code>	92, 94, 762, 763, 1018, 1019, 1173, 1174, 1330, 1331
<code>\gls@debug@nr</code>	7, 469, 487, 718, 737, 972, 1127, 1281	<code>\gls@glsdefs@labellist</code>	92, 92, 762, 1018, 1173, 1330
<code>\gls@debug@val</code>	7, 469, 487, 718, 737, 972, 1127, 1281	<code>\gls@gobbleopt</code>	80, 81, 505, 506, 755, 756, 1011, 1012, 1166, 1323
<code>\gls@defdocnewglossaryentry</code>	91, 93, 117, 511, 512, 526, 762, 763, 777, 1018, 1019, 1034, 1172, 1173, 1189, 1329, 1330, 1346	<code>\gls@grplabel</code>	300, 688, 941, 1125
<code>\gls@defglossaryentry</code>	91–94, 103, 103, 511, 512, 518, 762, 763, 769, 1018, 1019, 1025, 1172–1174, 1180, 1329–1331, 1337	<code>\gls@hypergrouprrun</code>	302, 302, 688, 941, 1125
<code>\gls@delimN</code>	266, 267, 1108, 1264, 1429	<code>\gls@if@measuring</code>	111, 112, 113, 1031, 1032, 1186, 1343
<code>\gls@delimR</code>	266, 266, 1108, 1264, 1429	<code>\gls@ifnotmeasuring</code>	112, 113, 114, 214, 524, 525, 775, 776, 1031–1033, 1088, 1186–1188, 1244, 1343–1345, 1401
<code>\gls@disablepagerefexpansion</code>	213, 214, 219, 581–583, 833–835, 1088, 1089, 1091, 1244, 1246, 1401, 1403	<code>\gls@inlinepostchild</code>	304, 305, 673, 689, 690, 926, 942, 943
<code>\gls@do@addxdyattribute</code>	64, 496, 746, 1002, 1157, 1314	<code>\gls@inlinesep</code>	304, 673, 689, 690, 926, 942, 943
<code>\gls@docdef@addlabel</code>	92, 93, 762, 763, 1018, 1019, 1173, 1330	<code>\gls@inlinesubsep</code>	304, 305, 673, 689, 690, 926, 942, 943
<code>\gls@docdef@labellist</code>	92, 92, 762, 1018, 1173, 1330	<code>\gls@islistofacronyms</code>	23, 476, 477, 726, 980, 981, 1135, 1290
<code>\gls@docclearpage</code>	60, 61, 494, 495, 744, 745, 1000, 1155, 1312	<code>\gls@istfilebase</code>	54, 55, 55, 200, 201, 492, 574, 575, 742, 826, 827, 997, 998, 1080, 1081, 1152, 1236, 1237, 1309, 1393, 1394
<code>\gls@doentryfmt</code>	79, 505, 755, 1011, 1166, 1322	<code>\gls@kernelglossredefs@nr</code>	44, 992, 1147, 1302
		<code>\gls@kernelglossredefs@val</code>	44, 992, 1147, 1302
		<code>\gls@label</code>	269, 603, 855, 1109, 1265, 1430

<code>\gls@level</code>	89, 105, 106, 244, 248, 510, 520, 593, 594, 760, 771, 845, 1016, 1027, 1101, 1171, 1181, 1182, 1256, 1257, 1328, 1338, 1339, 1418, 1421	<code>\gls@org@refstepcounter</code> . .	111, 112, 113, 1031, 1032, 1186, 1343
<code>\gls@local@stepcounter</code>	112, 112, 1031, 1186, 1343	<code>\gls@org@target</code>	112, 113, 1031, 1032, 1186, 1343
<code>\gls@measuring@reset</code> . .	112, 113, 1031, 1032, 1186, 1187, 1343, 1344	<code>\gls@orgAlph</code>	218, 219, 583, 835, 1090, 1091, 1246, 1403
<code>\gls@mfirstuc@nr</code>	45, 993, 994, 1148, 1304	<code>\gls@orgalph</code>	218, 219, 583, 835, 1090, 1091, 1246, 1403
<code>\gls@mfirstuc@val</code> 45, 993, 1148, 1304		<code>\gls@orgarabic</code>	218, 219, 583, 835, 1090, 1246, 1403
<code>\gls@newacronym</code>	269, 270, 1110, 1265, 1431	<code>\gls@orgnumber</code>	218, 583, 835, 1090, 1246, 1403
<code>\gls@noidx@nonumberlist</code> 86, 208,	1014, 1087, 1169, 1242, 1326, 1399	<code>\gls@orgRoman</code>	218, 219, 583, 835, 1090, 1091, 1246, 1403
<code>\gls@noidx@glossary</code>	206, 209, 577, 580, 829, 832, 1084, 1087, 1240, 1243, 1397, 1399	<code>\gls@orgromannumeral</code>	218, 583, 835, 1090, 1246, 1403
<code>\gls@nonumberlist@nr</code>	86, 508, 758, 1014, 1169, 1326	<code>\gls@orgthe</code>	218, 583, 835, 1090, 1246, 1403
<code>\gls@nonumberlist@val</code>	86, 508, 758, 1014, 1169, 1326	<code>\gls@original@glossary</code>	43, 44, 486, 487, 736, 737, 992, 1147, 1302, 1303
<code>\gls@nosetquote</code>	103, 193, 195, 198, 518, 569, 570, 573, 769, 820, 822, 825, 1025, 1075, 1077, 1079, 1180, 1231, 1232, 1235, 1337, 1388, 1389, 1392	<code>\gls@original@makeglossary</code>	43, 44, 486, 487, 736, 737, 992, 993, 1147, 1302, 1303
<code>\gls@number</code>	218, 583, 835, 1090, 1246, 1403	<code>\gls@printglossary@begin</code>	226, 227, 1250, 1251, 1407, 1408
<code>\gls@numberedsection@nr</code> 11, 253, 471, 595, 596, 721, 847, 975, 1103, 1129, 1258, 1259, 1284, 1424		<code>\gls@printglossary@end</code>	227, 227, 1251, 1408
<code>\gls@numberedsection@val</code>	11, 253, 471, 595, 721, 847, 975, 1103, 1129, 1258, 1284, 1424	<code>\gls@protected@pagefmts</code>	135, 214, 214, 215, 539, 582, 790, 833, 834, 1044, 1089, 1199, 1244, 1245, 1356, 1401, 1402
<code>\gls@numberpage</code>	214, 215, 218, 582, 583, 833–835, 1089, 1090, 1244–1246, 1401, 1403	<code>\gls@Romanpage</code>	214, 215, 219, 582, 583, 833–835, 1089, 1091, 1244–1246, 1401–1403
<code>\gls@org@glossaryentryfield</code>	226, 587, 839, 1095, 1250, 1407	<code>\gls@romanpage</code>	214, 215, 218, 582, 583, 833–835, 1089, 1090, 1244–1246, 1401–1403
<code>\gls@org@glossarysubentryfield</code>	226, 227, 587, 839, 1095, 1250, 1251, 1407	<code>\gls@save@numberlist</code> . . . 13, 223, 471, 472, 586, 721, 837, 975, 1093, 1130, 1249, 1284, 1285, 1406	
<code>\gls@org@insert</code>	620, 622, 624, 872, 874, 875, 877	<code>\gls@saveglossarygroups@nr</code>	40
<code>\gls@org@label</code>	111–113, 1031, 1032, 1186, 1343	<code>\gls@saveglossarygroups@val</code>	40
<code>\gls@org@link</code>	112, 113, 1031, 1032, 1186, 1343	<code>\gls@seen@index@nr</code>	9, 470, 720, 974, 1129, 1283
		<code>\gls@seen@index@val</code>	9, 470, 720, 974, 1129, 1283
		<code>\gls@set@xr@key</code>	85, 85, 508, 758, 1014, 1168, 1325

<code>\gls@start@measuring</code>	<code>\gls@xindy@glsnumberstrue</code>
. 112, 112, 113, 373,	. . . 36, 484, 734, 988, 1143, 1298
1031, 1032, 1186, 1187, 1343, 1344	<code>\gls@xr@key</code> 9, 10, 86, 86, 470, 471,
<code>\gls@stop@measuring</code> . . . 113, 113,	508, 720, 758, 974, 1014, 1129,
377, 1032, 1186, 1187, 1343, 1344	1168, 1169, 1283, 1284, 1325, 1326
<code>\gls@suffixF</code> 56, 56, 192, 194, 492,	<code>\glsaccessdisplay</code> 441, 640, 893
568, 570, 670, 672, 742, 820–	<code>\glsaccessibility</code>
822, 924, 925, 998, 1074–1076,	429, 437, 438, 632, 637, 885, 890
1153, 1230, 1232, 1310, 1387, 1389	<code>\glsaccsupp</code> 437, 438, 438, 637, 890, 891
<code>\gls@suffixFF</code> 56, 56, 192, 194, 492,	<code>\glsacronymtrue</code>
568, 570, 670–672, 742, 820,	. . . 22, 476, 726, 980, 1135, 1289
822, 924, 925, 998, 1074–1076,	<code>\glsacrpluralsuffix</code>
1153, 1230, 1232, 1310, 1387, 1389	51, 52, 270, 278–280, 284–286,
<code>\gls@text</code> 127, 535,	490, 603, 609, 612–614, 616,
786, 1040, 1194, 1195, 1351, 1352	740, 855, 861, 864–866, 869,
<code>\gls@the</code> 218, 583, 835, 1090, 1246, 1403	996, 1110, 1116, 1119–1121,
<code>\gls@thissty</code>	1150, 1265, 1271, 1272, 1275,
. . . 34, 483, 733, 987, 1142, 1297	1276, 1307, 1431, 1437, 1440–1442
<code>\gls@tmp</code> 202, 205, 211, 212,	<code>\glsacrshortcutsfalse</code>
291, 580, 624, 832, 877, 1082,	. . . 49, 489, 739, 994, 1149, 1306
1084, 1087, 1122, 1238, 1239,	<code>\glsacrshortcutstrue</code>
1243, 1278, 1395, 1396, 1400, 1443	. . . 49, 488, 739, 994, 1149, 1306
<code>\gls@tmplen</code>	<code>\glsacspace</code> 278, 278,
. 142, 143, 359, 416–418, 545,	281, 607, 608, 610, 859–861,
679, 715–717, 797, 932, 969,	1114–1116, 1270, 1272, 1435–1437
970, 1051, 1205, 1362, 1452–1454	<code>\glsadd</code> . . . 43, 186, 186, 187, 487,
<code>\gls@tr@set@acronym@toctitle</code> . .	564, 565, 737, 816, 992, 1071,
. 22, 476,	1147, 1226, 1227, 1302, 1383, 1384
725, 726, 980, 1134, 1135, 1289	<code>\glsaddall</code>
<code>\gls@tr@set@main@toctitle</code> 186, 565, 816, 1071, 1227, 1384
. . . 21, 475, 725, 979, 1134, 1289	<code>\glsaddallunused</code>
<code>\gls@tr@set@numbers@toctitle</code> . .	. 187, 565, 816, 1071, 1227, 1384
. . . 45, 488, 738, 993, 1148, 1303	<code>\glsaddkey</code>
<code>\gls@tr@set@symbols@toctitle</code> 96, 514, 765, 1021, 1176, 1333
. . . 44, 487, 738, 993, 1148, 1303	<code>\GlsAddLetterGroup</code>
<code>\gls@translate@nr</code> 70, 501, 751, 1007, 1161, 1318
. . . 33, 482, 732, 986, 1141, 1296	<code>\glsaddprotectedpagefmt</code> . . 215,
<code>\gls@translate@val</code>	215, 582, 834, 1089, 1245, 1402
. . . 33, 482, 732, 986, 1141, 1296	<code>\GlsAddSortRule</code> 68,
<code>\gls@wrglossary</code> 213,	499, 500, 750, 1005, 1160, 1317
213, 581, 833, 1088, 1244, 1401	<code>\glsaddstoragekey</code>
<code>\gls@xdy@set@lang</code> 95, 514, 764, 1020, 1175, 1332
. . . . 37, 200, 201, 227, 988,	<code>\GlsAddToMakeIndexOpts</code> . . . 195, 196
1080, 1081, 1095, 1143, 1236,	<code>\GlsAddXdyAlphabet</code>
1237, 1251, 1298, 1393, 1394, 1408	. . 65, 497, 747, 1003, 1158, 1315
<code>\gls@xdystring</code> 135, 539,	<code>\GlsAddXdyAttribute</code>
540, 790, 791, 1044, 1199, 1356 63, 63, 64, 65, 496,
<code>\gls@xindy@glsnumbersfalse</code>	497, 667, 746, 747, 920, 921,
. . . 37, 484, 734, 989, 1143, 1299	1001–1003, 1156–1158, 1313–1315

<code>\GlsAddXdyCounters</code>	62 , 63 , 64 , 82 , 495 , 496 , 506 , 667 , 745–747 , 756 , 920 , 1001 , 1002 , 1012 , 1156 , 1157 , 1167 , 1313 , 1314 , 1324	521 , 538 , 667 , 727 , 739 , 745 , 756 , 772 , 789 , 920 , 981 , 995 , 1001 , 1012 , 1028 , 1042 , 1136 , 1149 , 1156 , 1166 , 1183 , 1197 , 1290 , 1306 , 1313 , 1323 , 1340 , 1354
<code>\GlsAddXdyLocation</code>	67 , 498 , 499 , 667 , 749 , 921 , 1004 , 1005 , 1159 , 1160 , 1316	<code>\glscurrententrylabel</code>
<code>\GlsAddXdyStyle</code>	69 , 500 , 750 , 1006 , 1160 , 1317	143 , 223 , 226 , 227 , 586 , 587 , 838 , 839 , 1093 , 1095 , 1206 , 1249–1251 , 1363 , 1406 , 1407
<code>\GlsAppToWriteIstHook</code>	195	<code>\glscurrentfieldvalue</code>
<code>\glsclearpage</code>	41 , 202 , 203 , 485 , 486 , 576 , 736 , 828 , 991 , 1082 , 1083 , 1145 , 1146 , 1238 , 1301 , 1395	77 , 78 , 78 , 504 , 505 , 754 , 755 , 1010 , 1165 , 1321 , 1322
<code>\glsclearpage</code>	41 , 42 , 485 , 736 , 991 , 1145 , 1146 , 1301	<code>\glscustomtext</code>
<code>\glsclearpage</code>	11 , 11 , 253 , 254 , 471 , 596 , 721 , 848 , 975 , 1103 , 1129 , 1130 , 1259 , 1284 , 1424	122 , 124 , 126 , 145–150 , 166– 174 , 282 , 283 , 441 , 444 , 446– 449 , 459 , 460 , 529 , 532–534 , 546–549 , 555–559 , 611 , 616 , 617 , 620 , 622 , 624 , 640 , 643 , 645–648 , 656 , 780 , 782–785 , 798–800 , 806–811 , 863 , 869 , 872–875 , 877 , 893 , 896 , 898 , 900–902 , 909 , 910 , 1037–1039 , 1051–1054 , 1060–1065 , 1117 , 1118 , 1192–1194 , 1207–1209 , 1216–1220 , 1273 , 1274 , 1349– 1351 , 1364–1366 , 1373–1377 , 1439
<code>\glscapitalisewords</code>	180 , 180 , 383– 386 , 813 , 1067 , 1068 , 1223 , 1380	<code>\GlsDeclareNoHyperList</code>
<code>\glscapscase</code>	122 , 123 , 125 , 126 , 145–150 , 166–174 , 282 , 283 , 441 , 443– 449 , 459 , 460 , 529 , 530 , 532– 534 , 546–549 , 555–559 , 611 , 620 , 624 , 640 , 641 , 643–648 , 656 , 780 , 781 , 783–785 , 798– 800 , 806–811 , 863 , 873 , 877 , 893 , 894 , 896–902 , 909 , 1037– 1039 , 1051–1054 , 1060–1065 , 1118 , 1192–1194 , 1207–1209 , 1216–1220 , 1273 , 1274 , 1349– 1351 , 1364–1366 , 1373–1377 , 1439	24 , 24 , 477 , 727 , 981 , 1136 , 1290 , 1291
<code>\glscategory</code>	437 , 637 , 890	<code>\glndefaultshortaccess</code>
<code>\glsclearpage</code>	60 , 61 , 494 , 495 , 744 , 745 , 1000 , 1154 , 1155 , 1311 , 1312	465 , 465 , 661–664 , 914–917
<code>\glsclosebrace</code>	67 , 187 , 192 , 193 , 499 , 565 , 568 , 670 , 671 , 749 , 817 , 820 , 923 , 924 , 1004 , 1005 , 1071 , 1074 , 1075 , 1159 , 1227 , 1230 , 1316 , 1384 , 1387	<code>\glndefaulttype</code>
<code>\glscapitalisewords</code>	55 , 55 , 56 , 66 , 194 , 492 , 498 , 570 , 670 , 672 , 742 , 748 , 821 , 923 , 925 , 998 , 1004 , 1076 , 1152 , 1153 , 1159 , 1232 , 1309 , 1310 , 1316 , 1389	21 , 21 , 37 , 58 , 69 , 71 , 79 , 80 , 104 , 122 , 212 , 224 , 225 , 475 , 493 , 500 , 501 , 505 , 519 , 529 , 535 , 580 , 586 , 587 , 725 , 743 , 750 , 751 , 755 , 770 , 780 , 786 , 832 , 838 , 979 , 988 , 998 , 1006 , 1007 , 1011 , 1026 , 1037 , 1087 , 1094 , 1134 , 1143 , 1153 , 1161 , 1162 , 1165 , 1166 , 1181 , 1191 , 1192 , 1243 , 1249 , 1250 , 1289 , 1298 , 1310 , 1318 , 1319 , 1322 , 1323 , 1338 , 1348 , 1349 , 1400 , 1406 , 1407
<code>\glscapitalisewords</code>	180 , 180 , 383– 386 , 813 , 1067 , 1068 , 1223 , 1380	<code>\glndefmain</code>
<code>\glscapscase</code>	122 , 123 , 125 , 126 , 145–150 , 166–174 , 282 , 283 , 441 , 443– 449 , 459 , 460 , 529 , 530 , 532– 534 , 546–549 , 555–559 , 611 , 620 , 624 , 640 , 641 , 643–648 , 656 , 780 , 781 , 783–785 , 798– 800 , 806–811 , 863 , 873 , 877 , 893 , 894 , 896–902 , 909 , 1037– 1039 , 1051–1054 , 1060–1065 , 1118 , 1192–1194 , 1207–1209 , 1216–1220 , 1273 , 1274 , 1349– 1351 , 1364–1366 , 1373–1377 , 1439	21 , 21 , 82 , 475 , 506 , 725 , 756 , 979 , 1012 , 1134 , 1167 , 1289 , 1324
<code>\glscapitalisewords</code>	180 , 180 , 383– 386 , 813 , 1067 , 1068 , 1223 , 1380	<code>\glsdefs@newdocentry</code>
<code>\glscapitalisewords</code>	180 , 180 , 383– 386 , 813 , 1067 , 1068 , 1223 , 1380	93 , 94 , 763 , 1019 , 1174 , 1331
<code>\glscapitalisewords</code>	180 , 180 , 383– 386 , 813 , 1067 , 1068 , 1223 , 1380	<code>\GLSdesc</code>
<code>\glscapitalisewords</code>	180 , 180 , 383– 386 , 813 , 1067 , 1068 , 1223 , 1380	157 , 551 , 803 , 1056 , 1212 , 1369

<code>\Glsdesc</code>	156,	1220, 1221, 1225, 1226, 1240–
	156 , 551 , 803 , 1056 , 1212 , 1369	1242 , 1246 , 1250–1253 , 1256–
<code>\glsdesc</code>	156,	1258 , 1260 , 1261 , 1278–1280 ,
	156 , 551 , 803 , 1056 , 1212 , 1369	1300 , 1319–1322 , 1326 , 1331 ,
<code>\GLSdescplural</code>		1334–1337 , 1341 , 1345–1347 ,
	. 158 , 552 , 803 , 1057 , 1212 , 1369	1354 , 1377 , 1378 , 1382 , 1383 ,
<code>\Glsdescplural</code>	157,	1397–1399 , 1402 , 1403 , 1407 ,
	157 , 551 , 803 , 1057 , 1212 , 1369	1414 , 1415 , 1417–1419 , 1421–
<code>\glsdescplural</code>	157, 157 ,	1423 , 1425 , 1426 , 1444 , 1445 , 1453
	551 , 803 , 1056 , 1057 , 1212 , 1369	<code>\glsdisablehyper</code>
<code>\glsdescriptionaccessdisplay</code> ..		. 144 , 546 , 797 , 1051 , 1207 , 1364
 439 , 451 ,	<code>\Glsdisp</code> .. 151 , 151 , 1054 , 1209 , 1366
	455 , 638 , 641–643 , 650 , 652 ,	<code>\glsdisp</code>
	653 , 892 , 895 , 896 , 903 , 905 , 906	150 ,
<code>\glsdescriptionpluralaccessdisplay</code>		151 , 549 , 800 , 1054 , 1209 , 1366
 439 , 451 ,	<code>\glsdisplay</code> 127 , 529 , 535 , 536 , 780 , 786
	638 , 640 , 641 , 650 , 892–894 , 903	<code>\glsdisplayfirst</code>
<code>\glsdescwidth</code>	310 , 310 , 311– 127 , 529 , 535 , 780 , 786
	313 , 315 , 316 , 318–321 , 321 ,	<code>\glsdisplaynumberlist</code>
	322–326 , 335 , 335 , 336–338 ,	184 , 207 , 563 , 578 , 815 , 830 ,
	340–342 , 342 , 343–348 , 692–	1070 , 1086 , 1225 , 1241 , 1382 , 1398
	695 , 697–701 , 705–712 , 946 ,	<code>\glsdohyperlink</code>
	947 , 949–955 , 958–960 , 962–966	144 , 144 , 145 , 545 , 546 , 797 ,
<code>\glsdetoklabel</code>		798 , 1051 , 1206 , 1207 , 1363 , 1364
	39 , 73 , 73 , 74–78 , 87 , 94 , 99–	<code>\glsdohyperlinkhook</code>
	103 , 109 , 114 , 115 , 117–119 , 144 , 144 , 1206 , 1363
	132 , 174–176 , 184–186 , 206–	<code>\glsdohypertarget</code>
	208 , 216 , 219 , 226 , 227 , 237–	142 , 144 , 145 , 545 , 546 , 797 ,
	239 , 244 , 245 , 248 , 250 , 251 ,	798 , 1051 , 1205 , 1207 , 1362 , 1364
	256 , 258 , 293–297 , 349 , 416 ,	<code>\glsdohypertargethook</code>
	422 , 423 , 438–441 , 466 , 467 ,	143 , 143 , 1205 , 1206 , 1362 , 1363
	501–504 , 509 , 512 , 516–518 ,	<code>\glsdoifexists</code>
	523 , 525–527 , 537 , 559 , 560 ,	73 , 75 , 77 , 99–
	563 , 564 , 577–579 , 583 , 587 ,	102 , 113 , 114 , 145–151 , 166–
	589 , 590 , 593–595 , 597 , 598 ,	174 , 180 , 184 , 186 , 207 , 208 ,
	629–632 , 638–640 , 665 , 666 ,	230 , 424–428 , 447–449 , 454 ,
	684 , 716 , 735 , 752–754 , 759 ,	455 , 502–504 , 516–518 , 524 ,
	763 , 767–769 , 773 , 776–778 ,	525 , 546–549 , 555–559 , 561 ,
	788 , 811 , 815 , 816 , 829–831 ,	563 , 564 , 578 , 579 , 647 , 648 ,
	834 , 835 , 839 , 841 , 842 , 845–	652 , 685–687 , 752–754 , 767–
	847 , 849 , 850 , 882–885 , 891–	769 , 775 , 776 , 798–801 , 806–
	893 , 918 , 919 , 937 , 938 , 969 ,	810 , 813 , 815 , 816 , 830 , 831 ,
	990 , 1007–1010 , 1014 , 1015 ,	900 , 901 , 905 , 906 , 938–941 ,
	1019 , 1023–1025 , 1030 , 1033–	1007–1010 , 1023–1025 , 1032 ,
	1035 , 1042 , 1065 , 1066 , 1070 ,	1033 , 1051–1054 , 1060–1065 ,
	1071 , 1085 , 1086 , 1090 , 1091 ,	1068–1071 , 1085 , 1086 , 1162–
	1095 , 1097 , 1098 , 1101 , 1102 ,	1164 , 1177–1179 , 1187 , 1188 ,
	1104 , 1105 , 1123 , 1124 , 1145 ,	1207–1210 , 1216–1220 , 1223 ,
	1162–1165 , 1169 , 1174 , 1177–	1225 , 1226 , 1241 , 1242 , 1319–
	1180 , 1184 , 1188–1190 , 1197 ,	1321 , 1334–1336 , 1344 , 1345 ,
		1364–1367 , 1373–1377 , 1380 ,
		1382 , 1383 , 1398 , 1399 , 1409

<code>\glsdoifexistsordo</code>	<code>\Glsentrydesc</code>
130, 175, 502, 537, 559, 752,	157, <u>177</u> ,
787, 811, 1008, 1041, 1065,	178, 259, 451, 455, 551, 560,
1163, 1196, 1221, 1319, 1353, 1378	599, 650, 652, 803, 812, 851,
<code>\glsdoifexistsorwarn</code>	903, 905, 1056, 1066, 1106,
74,	1212, 1222, 1262, 1369, 1379, 1427
250, 258–260, 502, 595, 598,	<code>\glsentrydesc</code>
599, 752, 847, 850, 851, 1008,	143,
1102, 1105, 1106, 1162, 1258,	156, 157, <u>177</u> , 259, 451, 455,
1261, 1262, 1319, 1422, 1426, 1427	530–532, 551, 560, 599, 641–
<code>\glsdoifnoexists</code>	643, 650, 652, 781–783, 803,
73,	812, 850, 895, 896, 903, 905,
91, 102, 502, 511, 518, 752,	1056, 1066, 1106, 1206, 1212,
761, 769, 1008, 1017, 1025,	1222, 1261, 1363, 1369, 1379, 1427
1162, 1172, 1180, 1319, 1329, 1337	<code>\glsentrydescaccess</code>
<code>\glsdonohyperlink</code> <u>435</u> , 439, 636, 638, 889, 892
.	<code>\Glsentrydescplural</code>
132, 144, <u>144</u> , 538,
545, 546, 789, 797, 1042, 1051,	158, 178, <u>178</u> , 451,
1197, 1206, 1207, 1354, 1363, 1364	552, 561, 650, 803, 812, 903,
<code>\glsdoparenifnotempty</code>	1057, 1067, 1212, 1222, 1369, 1379
. <u>291</u> , 624, 876, 1122, 1278, 1443	<code>\glsentrydescplural</code>
<code>\glsdosanitizesort</code>	157,
17, 473,	158, <u>178</u> , 451, 529, 530, 551,
474, 723, 977, 1132, 1286, 1287	552, 560, 640, 641, 650, 780,
<code>\glsdoshowtarget</code>	781, 803, 812, 893, 894, 903,
.	1057, 1067, 1212, 1222, 1369, 1379
8, 142, 144, 974, 1051,	<code>\glsentrydescpluralaccess</code>
1128, 1205, 1206, 1283, 1362, 1363 <u>435</u> , 439, 636, 639, 889, 892
<code>\glsenableentrycount</code>	<code>\Glsentryfirst</code>
. . . . <u>116</u> , 526, 527, 777, 778,	120, 124, 153, 179, <u>179</u> , 443,
1034, 1035, 1189, 1190, 1346, 1347	450, 528, 531, 533, 550, 561,
<code>\glsenablehyper</code>	642, 644, 649, 779, 782, 784,
. <u>145</u> , 546, 797, 1051, 1207, 1364	801, 813, 895, 898, 902, 1036,
<code>\glsencapwrcontent</code>	1038, 1055, 1067, 1191, 1193,
.	1210, 1223, 1348, 1350, 1367, 1380
132, <u>133</u> , 1042, 1197, 1354	<code>\glsentryfirst</code> 119, 124, 152, 153,
<code>\glsentryaccess</code>	<u>179</u> , 443, 444, 450, 528, 530,
. . . . <u>434</u> , 438, 636, 638, 889, 891	531, 533, 550, 561, 642, 644,
<code>\glsentrycounter</code>	645, 649, 779, 781, 782, 784,
133,	801, 813, 895–898, 902, 1036,
264, 267, 538, 600, 602, 789,	1038, 1055, 1067, 1191–1193,
852, 854, 1042, 1107, 1109,	1210, 1223, 1348–1350, 1367, 1380
1197, 1263, 1264, 1354, 1428, 1430	<code>\glsentryfirstaccess</code>
<code>\glsentrycounterfalse</code> <u>434</u> , 439, 636, 638, 889, 891
. . . 16, 473, 723, 977, 1131, 1286	<code>\Glsentryfirstplural</code>
<code>\glsentrycounterlabel</code> 121, 123, 155, <u>180</u> , <u>180</u> ,
.	442, 450, 529, 530, 532, 550,
257, <u>257</u> , 597, 598,	561, 641, 644, 649, 780, 781,
849, 850, 1105, 1260, 1261, 1426	783, 802, 813, 894, 897, 902,
<code>\GlsEntryCounterLabelPrefix</code> . . .	1037, 1056, 1067, 1191, 1192,
. . .	1211, 1223, 1348, 1349, 1368, 1380
<u>254</u> , 256, 596, 597, 848, 849,	<code>\glsentryfirstplural</code>
1103–1105, 1259, 1260, 1424–1426	121,
<code>\glsentrycountertrue</code>	123, 154, 155, <u>179</u> , 442, 450,
. . . 16, 473, 723, 977, 1131, 1286	
<code>\glsentrycurrcount</code>	
. 117–119, 526–528, <u>777</u> – <u>779</u> ,	
1034–1036, 1189, 1190, 1346, 1347	

528–530, 532, 550, 551, 561, 640, 641, 643, 644, 649, 779– 781, 783, 802, 813, 893, 894, 896, 897, 902, 903, 1036, 1037, 1055, 1056, 1067, 1191, 1192, 1211, 1223, 1348, 1349, 1368, 1380	809, 811, 814, 815, 859, 863, 864, 900, 901, 906, 909, 910, 1036, 1063, 1066, 1069, 1114, 1118, 1119, 1191, 1219, 1222, 1225, 1269, 1270, 1274, 1348, 1376, 1379, 1382, 1435, 1439, 1440
<code>\glsentryfirstpluralaccess</code> <u>435</u> , 439, 636, 638, 889, 891	<code>\glsentrylong</code> 119, 126, 170, 172, 177, 183, <u>183</u> , 277–287, 446, 448, 449, 456–465, 528, 535, 557, 558, 560, 563, 607–615, 622, 646, 648, 653–660, 779, 785, 809, 811, 814, 815, 859–867, 875, 900–902, 906–914, 1036, 1040, 1063, 1066, 1069, 1114–1121, 1191, 1194, 1218, 1219, 1221, 1225, 1269–1277, 1348, 1351, 1375, 1376, 1378, 1382, 1435–1442
<code>\glsentryfmt</code> 81, 83, <u>122</u> , 506, 529, 756, 757, 780, 1012, 1037, 1166, 1167, 1192, 1323, 1324, 1349	<code>\glsentrylongaccess</code> <u>435</u> , 440, 636, 639, 889, 892
<code>\Glsentryfull</code> 183, <u>183</u> , 275, 284, 286, 461, 463, 563, 606, 612, 613, 657, 659, 815, 858, 864, 865, 910, 912, 1069, 1112, 1119, 1120, 1225, 1268, 1274, 1276, 1382, 1434, 1440, 1441	<code>\Glsentrylongpl</code> 121, 173, <u>183</u> , <u>183</u> , 184, 277, 278, 283, 284, 447, 456, 459, 461, 529, 559, 563, 607, 608, 611, 612, 647, 653, 656, 657, 780, 810, 814, 815, 859, 860, 863, 864, 900, 906, 909–911, 1037, 1064, 1069, 1114, 1118, 1119, 1191, 1220, 1225, 1269, 1270, 1273–1275, 1348, 1377, 1382, 1435, 1439, 1440
<code>\glsentryfull</code> <u>183</u> , 275, 284, 286, 461, 463, 563, 606, 612, 613, 657, 659, 814, 858, 864, 865, 910, 912, 1069, 1112, 1119, 1120, 1225, 1268, 1274, 1276, 1382, 1434, 1440, 1441	<code>\glsentrylongpl</code> 121, 127, 172, 174, 183, <u>183</u> , 277–279, 282–286, 447, 456, 457, 459, 461–463, 528, 535, 558, 559, 563, 607, 608, 611–614, 622, 628, 647, 653, 654, 656–659, 779, 786, 810, 811, 814, 815, 859, 860, 863–865, 875, 881, 900, 906, 907, 909–912, 1036, 1040, 1064, 1065, 1069, 1114, 1115, 1118–1120, 1191, 1195, 1219, 1220, 1225, 1269–1271, 1273–1276, 1348, 1352, 1376, 1377, 1382, 1435, 1436, 1439–1441
<code>\Glsentryfullpl</code> 184, <u>184</u> , 275, 284, 286, 461, 463, 563, 606, 612, 614, 657, 659, 815, 858, 864, 865, 911, 912, 1069, 1113, 1119, 1120, 1225, 1268, 1275, 1276, 1382, 1434, 1440, 1441	<code>\glsentrylongpluralaccess</code> <u>436</u> , 440, 636, 639, 890, 892
<code>\glsentryfullpl</code> <u>183</u> , 275, 284, 286, 461, 463, 563, 606, 612, 614, 657, 659, 815, 858, 864, 865, 910, 912, 1069, 1113, 1119, 1120, 1225, 1268, 1274, 1276, 1382, 1434, 1440, 1441	<code>\Glsentryname</code> 156, 176, <u>176</u> , 258, 451, 455, 551, 560, 598, 650, 652, 802, 811, 850, 903, 905, 1056, 1065, 1066, 1106,
<code>\glsentryitem</code> <u>257</u> , 304, 306–312, 314, 322, 323, 325, 335, 337, 339, 342, 344, 346, 406, 411, 413, 414, 417, 598, 673–678, 680–682, 689, 691–694, 698–700, 705–707, 709–711, 713–716, 850, 926– 935, 942, 944–948, 952–954, 959–961, 963–965, 967, 968, 970, 1105, 1261, 1426, 1450–1453	
<code>\Glsentrylong</code> 120, 171, 177, 183, <u>183</u> , 277, 278, 283, 284, 446, 449, 456, 460, 461, 528, 557, 560, 563, 607, 611, 612, 647, 648, 653, 656, 657, 779,	

1212, 1221, 1261, 1369, 1378, 1427	\Glsentryshort 125, 167, 177, 182,
\glsentryname 143, 155, 156,	182, 279, 285, 286, 445–447,
175, 223, 259, 349, 416, 451,	457, 462, 463, 534, 555, 560,
454, 551, 560, 649, 650, 652,	562, 608, 613, 646, 647, 654,
716, 802, 803, 811, 903, 905,	658, 659, 785, 807, 812, 814,
969, 1056, 1065, 1066, 1093,	860, 865, 899, 900, 907, 911,
1206, 1211, 1212, 1221, 1249,	912, 1039, 1061, 1066, 1069,
1363, 1368, 1369, 1378, 1406, 1453	1115, 1119, 1120, 1194, 1216,
\glsentrynumberlist . . . 184, 184,	1222, 1224, 1271, 1275, 1276,
207, 563, 578, 815, 830, 1069,	1351, 1373, 1379, 1381, 1436, 1441
1070, 1085, 1225, 1241, 1382, 1398	\glsentryshort 125, 127,
\glsentryparent	166, 168, 177, 182, 183, 276–
. 180, 562, 813, 1068, 1223, 1380	287, 445–448, 456–465, 534,
\Glsentryplural	535, 555, 556, 560, 562, 563,
123, 154, 178, 178, 442, 450,	606–615, 645–648, 653–660,
530, 532, 550, 561, 640, 643,	785, 806, 807, 812, 814, 815,
649, 780, 783, 802, 812, 894,	858–867, 899–901, 906–914,
897, 902, 1037, 1055, 1067,	1039, 1040, 1060, 1061, 1066,
1192, 1211, 1222, 1349, 1368, 1379	1069, 1113–1121, 1193, 1194,
\glsentryplural	1216, 1217, 1222, 1224, 1225,
123, 153, 154, 178, 442, 450,	1268–1277, 1350, 1351, 1373,
529, 530, 532, 550, 561, 640,	1374, 1379, 1381, 1382, 1434–1442
641, 643, 644, 649, 780, 781,	\glsentryshortaccess
783, 802, 812, 893, 894, 896, 435, 440, 636, 639, 889, 892
897, 902, 1037, 1055, 1067,	\Glsentryshortpl 125, 169,
1192, 1211, 1222, 1349, 1368, 1379	183, 183, 279, 285, 286, 444,
\glsentrypluralaccess	457, 462, 463, 533, 556, 563,
. . . . 435, 439, 636, 638, 889, 891	608, 613, 614, 645, 654, 658,
\Glsentryprefix	659, 784, 808, 814, 860, 865,
423, 423, 425, 684, 686, 937, 939	898, 907, 911, 912, 1038, 1062,
\glsentryprefix . . 422, 423, 424,	1069, 1115, 1120, 1193, 1218,
427, 684, 685, 687, 937, 938, 940	1224, 1225, 1271, 1275, 1276,
\Glsentryprefixfirst	1350, 1375, 1381, 1382, 1436, 1441
422, 422, 426, 684, 686, 937, 939	\glsentryshortpl 125, 127, 168, 170,
\glsentryprefixfirst 422, 422, 424,	182, 183, 184, 277–279, 284–
427, 684, 685, 687, 937, 938, 940	286, 444, 445, 447, 456, 457,
\Glsentryprefixfirstplural	461–463, 533, 535, 556, 557,
422, 422, 426, 684, 687, 937, 940	563, 607, 608, 612–614, 628,
\glsentryprefixfirstplural	645, 647, 653, 654, 657–659,
. 422, 422, 425,	784, 786, 808, 814, 815, 859,
428, 684, 685, 688, 937, 939, 941	860, 864, 865, 881, 898, 900,
\Glsentryprefixplural	906, 907, 910–912, 1038, 1040,
423, 423, 426, 684, 686, 937, 940	1062, 1069, 1114, 1115, 1119,
\glsentryprefixplural	1120, 1193, 1195, 1217, 1218,
. 422, 423, 425,	1224, 1225, 1269–1271, 1274–
428, 684, 685, 687, 937, 939, 941	1276, 1350, 1352, 1374, 1375,
\glsentryprevcount	1381, 1382, 1435, 1436, 1440, 1441
117, 118, 526, 527, 777, 778,	\glsentryshortpluralaccess
1034, 1035, 1189, 1190, 1346, 1347 435, 440, 636, 639, 889, 892
	\glsentrysort

. 180, 562, 813, 1068, 1223, 1380

\Glsentrysymbol 159, 179,
179, 260, 452, 455, 552, 561,
599, 650, 653, 804, 812, 851,
903, 906, 1057, 1067, 1106,
1213, 1223, 1262, 1370, 1380, 1427

\glsentrysymbol .. 143, 158, 159,
179, 260, 349, 452, 455, 530–
532, 552, 561, 599, 617, 620,
624, 641–643, 650, 652, 781–
783, 803, 804, 812, 851, 869,
873, 877, 895, 896, 903, 906,
1057, 1067, 1106, 1206, 1213,
1222, 1262, 1363, 1370, 1379, 1427

\glsentrysymbolaccess
. . . . 435, 439, 636, 638, 889, 891

\Glsentrysymbolplural
. 160, 179, 179, 452,
552, 561, 650, 804, 813, 904,
1058, 1067, 1213, 1223, 1370, 1380

\glsentrysymbolplural 159,
160, 179, 452, 529, 530, 552,
561, 617, 620, 624, 640, 641,
650, 780, 781, 804, 813, 869,
873, 877, 893, 894, 904, 1057,
1058, 1067, 1213, 1223, 1370, 1380

\glsentrysymbolpluralaccess . . .
. . . . 435, 439, 636, 638, 889, 892

\Glsentrytext
124, 152, 178, 178, 443, 449,
531, 533, 549, 561, 642, 644,
649, 782, 784, 801, 812, 895,
897, 902, 1038, 1054, 1067,
1193, 1210, 1222, 1350, 1367, 1379

\glsentrytext 123, 124,
151, 152, 178, 185, 223, 443,
449, 530, 531, 533, 549, 561,
564, 586, 641, 642, 644, 645,
649, 781–784, 801, 812, 816,
837, 895, 897, 898, 902, 1038,
1054, 1067, 1070, 1093, 1192,
1193, 1210, 1222, 1226, 1249,
1349, 1350, 1367, 1379, 1383, 1406

\glsentrytextaccess
. . . . 434, 438, 636, 638, 889, 891

\glsentrytitlecase 180, 259, 260,
561, 562, 813, 1068, 1223, 1380

\glsentrytype 104,
180, 252, 519, 562, 770, 813,
1026, 1068, 1181, 1223, 1338, 1380

\Glsentryuseri 161, 181, 181, 452,
553, 562, 651, 804, 814, 904,
1058, 1068, 1214, 1224, 1371, 1381

\glsentryuseri 160,
161, 181, 452, 552, 553, 562,
651, 804, 813, 904, 1058, 1068,
1213, 1214, 1223, 1370, 1371, 1380

\glsentryuseriaccess
. . . . 436, 440, 636, 639, 890, 892

\Glsentryuserii 162, 181, 181, 453,
553, 562, 651, 805, 814, 904,
1058, 1068, 1214, 1224, 1371, 1381

\glsentryuserii
. . . . 161, 162, 181, 453, 553,
562, 651, 804, 805, 814, 904,
1058, 1068, 1214, 1224, 1371, 1381

\glsentryuseriiaccess
. . . . 436, 440, 636, 639, 890, 892

\Glsentryuseriii 163, 181, 181, 453,
553, 562, 651, 805, 814, 904,
1059, 1068, 1214, 1224, 1371, 1381

\glsentryuseriii 162, 163, 181, 453,
553, 562, 651, 805, 814, 904,
1059, 1068, 1214, 1224, 1371, 1381

\glsentryuseriiiaccess
. . . . 436, 440, 637, 639, 890, 892

\Glsentryuseriv 164, 182, 182, 453,
554, 562, 651, 805, 814, 904,
1059, 1068, 1215, 1224, 1372, 1381

\glsentryuseriv
. . . . 163, 164, 181, 453, 554,
562, 651, 805, 814, 904, 905,
1059, 1068, 1215, 1224, 1372, 1381

\glsentryuserivaccess
. . . . 436, 441, 637, 639, 890, 893

\Glsentryuserv 165, 182, 182, 454,
554, 562, 651, 806, 814, 905,
1060, 1069, 1215, 1224, 1372, 1381

\glsentryuserv 164,
165, 182, 454, 554, 562, 651,
652, 806, 814, 905, 1059, 1060,
1068, 1069, 1215, 1224, 1372, 1381

\glsentryuservaccess
. . . . 436, 441, 637, 639, 890, 893

\Glsentryuservi 166, 182, 182, 454,
554, 562, 652, 806, 814, 905,
1060, 1069, 1216, 1224, 1373, 1381

\glsentryuservi 165,
166, 182, 454, 554, 555, 562,

652, 806, 814, 905, 1060, 1069, 1215, 1216, 1224, 1372, 1373, 1381	864, 898, 906, 907, 911, 1038, 1113–1115, 1119, 1193, 1269, 1270, 1275, 1350, 1435, 1436, 1440
<code>\glsentryuserviaccess</code>	<code>\glsentryuserviaccess</code>
. 436, 441, 637, 640, 890, 893 122,
<code>\glsesclocationsfalse</code>	122, 277–279, 283, 285, 289,
. 13, 206, 577, 829,	441, 456, 460, 462, 532, 607,
976, 1084, 1130, 1240, 1285, 1397	608, 611, 613, 615, 617, 618,
<code>\glsesclocationstrue</code> 472, 721	620, 622, 624, 626, 628, 643,
<code>\glsexpandfields</code>	653, 654, 656, 658, 783, 859,
. 91, 511, 761, 1017, 1172, 1329	860, 863, 864, 867, 869, 871,
<code>\glsfieldaccsupp</code>	872, 875, 877, 879, 881, 896,
437, 437, 438, 637, 638, 890, 891	906, 907, 910, 911, 1037, 1113–
<code>\glsfielddef</code>	1115, 1118, 1119, 1122, 1192,
. 100, 517, 767, 1023, 1178, 1335	1269, 1270, 1274, 1275, 1278,
<code>\glsfieldedef</code>	1349, 1435, 1436, 1439, 1440, 1443
. 99, 516, 767, 1023, 1178, 1335	<code>\glsgetgrouplabel</code>
<code>\glsfieldfetch</code> 263, 600, 852, 1107, 1262, 1428
100, 180, 517, 561, 768, 813,	<code>\glsgetgrouptitle</code>
1024, 1068, 1178, 1223, 1335, 1380 262, 303, 306–309,
<code>\glsfieldgdef</code>	329–334, 412–415, 419, 600,
. 99, 516, 767, 1023, 1178, 1335	689, 691, 692, 702–705, 713–
<code>\glsfieldxdef</code>	715, 717, 852, 942, 944, 945,
. 99, 516, 767, 1023, 1177, 1334	955–958, 967–969, 971, 1106,
<code>\glsfindwidesttoplevelname</code>	1126, 1262, 1427, 1446–1452, 1454
. 415, 715, 969, 1452	<code>\gls glossarymark</code> 59, 59, 493,
<code>\GLSfirst</code>	743, 999, 1153, 1154, 1310, 1311
. 153, 550, 801, 1055, 1210, 1367	<code>\gls groupheading</code> 193, 245, 249,
<code>\Glsfirst</code> 152,	261, 304, 307–310, 312, 314,
152, 550, 801, 1055, 1210, 1367	322, 323, 325, 329–335, 337,
<code>\glsfirst</code> 152, 152,	339, 342, 344, 346, 402, 411–
550, 801, 1054, 1055, 1210, 1367	415, 417–419, 568, 594, 600,
<code>\glsfirstaccessdisplay</code>	671, 689, 691–694, 698–700,
439, 443, 444, 450, 638, 642,	702–707, 709–711, 713–717,
644, 645, 649, 891, 895–898, 902	820, 846, 852, 924, 942, 944–
<code>\GLSfirstplural</code>	948, 952–961, 963–971, 1075,
. 155, 550, 802, 1056, 1211, 1368	1101, 1106, 1230, 1257, 1262,
<code>\Glsfirstplural</code> 154,	1387, 1419, 1421, 1427, 1446–1454
154, 550, 802, 1055, 1211, 1368	<code>\gls groupskip</code> 192, 193,
<code>\glsfirstplural</code> 154,	245, 249, 261, 305, 307, 311,
154, 550, 802, 1055, 1211, 1368	312, 314, 317, 318, 322, 324,
<code>\glsfirstpluralaccessdisplay</code>	325, 336, 337, 339, 343, 345,
. 439, 442, 450,	347, 350, 402, 412, 413, 415,
638, 640, 641, 643, 644, 649,	418, 568, 594, 600, 670, 671,
891, 893, 894, 896, 897, 902, 903	690, 691, 693, 694, 696, 699,
<code>\glsfirstpluralacessdisplay</code>	700, 705–707, 709–711, 713–
. 442, 644, 897	715, 717, 820, 846, 852, 923,
<code>\gls genacfmt</code>	924, 943, 944, 946–950, 952–
. 124, 277–279, 285, 444, 456,	954, 959–961, 963–965, 967–
462, 533, 607, 608, 613, 645,	969, 971, 1074, 1075, 1101,
653, 654, 658, 784, 859, 860,	1106, 1230, 1257, 1262, 1387,

1419, 1421, 1427, 1450–1452, 1454
`\glshyperfirstfalse` 285, 462, 613,
 658, 865, 911, 1119, 1275, 1440
`\glshyperfirsttrue`
 ... 34, 483, 733, 987, 1142, 1297
`\glshyperlink`
 185, 223, 564, 586, 816, 837,
 1070, 1093, 1226, 1249, 1383, 1406
`\glshypernavsep`
 303, 305, 689, 942, 1126
`\glshypernumber`
 . 57, 266, 268, 492, 601–603,
 743, 853–855, 998, 1108, 1109,
 1153, 1263, 1265, 1310, 1429, 1430
`\glshyper` 129, 536, 787
`\glshyperon` 131, 536,
 537, 787, 788, 1042, 1196, 1353
`\glsIfListOfAcronyms` 23, 23,
 476, 477, 726, 980, 981, 1135, 1290
`\glsifmeasuring` 112,
 112, 113, 1031, 1032, 1186, 1343
`\glsifplural`
 122, 125, 145–150, 166–
 174, 282, 441, 444, 445, 447–
 449, 459, 529, 532–534, 546–
 549, 555–559, 611, 617, 620,
 622, 624, 640, 643, 645–648,
 656, 780, 783–785, 798–800,
 806–811, 863, 869, 872, 875,
 877, 893, 896, 898–902, 909,
 1037–1039, 1051–1054, 1060–
 1065, 1118, 1192–1194, 1207–
 1209, 1216–1220, 1273, 1349–
 1351, 1364–1366, 1373–1377, 1439
`\glsifusedtranslator`dict
 ... 32, 482, 732, 986, 1141, 1296
`\glsifusetranslator` 32–34, 53, 419,
 420, 482, 483, 490, 491, 666,
 683, 731–733, 740, 741, 919,
 936, 986, 987, 996, 997, 1140–
 1142, 1151, 1152, 1296, 1297, 1308
`\glsignore`
 . 187, 565, 816, 1071, 1227, 1384
`\glsindexingsetting`
 16, 20, 203, 205, 977,
 979, 1083, 1084, 1132–1134,
 1238, 1240, 1286, 1288, 1395, 1397
`\glsindexonlyfirstfalse`
 ... 34, 483, 733, 987, 1142, 1297
`\glsinlinedescformat`
 304, 305, 673, 690, 926, 943
`\glsinlinedopostchild` 304,
 305, 673, 689, 690, 926, 942, 943
`\glsinlineemptydescformat`
 304, 305, 673, 689, 690, 926, 943
`\glsinlineifhaschildren` .. 304, 305
`\glsinlinenameformat`
 304, 305, 673, 689, 690, 926, 943
`\glsinlineparentchildseparator`
 304, 305, 673, 690, 926, 943
`\glsinlinepostchild`
 304, 305, 673, 690, 926, 943
`\glsinlineseparator`
 304, 305, 673, 690, 926, 943
`\glsinlinesubdescformat`
 305, 305, 673, 690, 926, 943, 944
`\glsinlinesubnameformat`
 304, 305, 673, 690, 926, 943, 944
`\glsinlinesubseparator`
 305, 305, 673, 690, 926, 943
`\glsinsert`
 . 123–126, 145–150, 166–174,
 282, 283, 442–449, 459, 460,
 529–534, 546–549, 555–559,
 611, 617, 620, 622, 624, 640–
 648, 656, 780–785, 798–800,
 806–811, 863, 869, 872–875,
 877, 893–902, 909, 910, 1037–
 1039, 1052–1054, 1060–1065,
 1118, 1192–1194, 1207–1209,
 1216–1220, 1273, 1274, 1349–
 1351, 1364–1366, 1373–1377, 1439
`\glskeylisttok` 274, 275, 465, 605,
 616–619, 621, 623, 625–629,
 661–665, 857, 867, 868, 870–
 876, 878–880, 882, 914–918,
 1111, 1112, 1267, 1268, 1432, 1433
`\glslabel` 103, 122–126,
 131, 132, 167–174, 277–279,
 282, 283, 285, 442–446, 456,
 459, 460, 462, 519, 529–534,
 537, 538, 555–559, 607, 608,
 611, 613, 617, 620, 622, 624,
 640–646, 653, 654, 656, 658,
 769, 780–785, 788, 789, 807–
 811, 859, 860, 863, 864, 869,
 872, 873, 875, 877, 893–899,
 906, 907, 909, 911, 1025, 1037–
 1039, 1041, 1042, 1061–1065,

1113–1115, 1118, 1119, 1180, 1192–1194, 1196, 1197, 1216– 1220, 1269, 1270, 1273–1275, 1337, 1349–1351, 1353, 1354, 1373–1377, 1435, 1436, 1439, 1440	549, 776, 777, 798–800, 1033, 1034, 1040, 1188, 1195, 1345, 1352
\glslabelhypertarget 143 , 1206, 1363	\glslocalunsetall 116 , 526, 777, 1034, 1188, 1345
\glslabelhypertargetdefs 143 , 143 , 1206, 1363	\glslocationncstoencap 57 , 998, 1153, 1310
\glslabelhypertargetprefix 143 , 143 , 1206, 1363	\glslongaccessdisplay 440 , 446, 448, 449, 456–465, 639, 646–648, 653–661, 892, 900–902, 906–914
\glslabelhypertargetvalue 143 , 143 , 1206, 1363	\glslongaccesskey 465 , 665, 918
\glslabeltok 274 , 275, 605, 615–629, 661–664, 857, 867–877, 879–882, 914– 917, 1111, 1112, 1267, 1268, 1433	\glslongkey 270 , 465, 603, 665, 855, 918, 1110, 1266, 1431
\glsletentryfield 174 , 559, 811, 1065, 1220, 1377	\glslongpluralaccessdisplay 440 , 447, 456, 457, 459, 461–463, 639, 647, 653, 654, 656–659, 661, 892, 900, 906, 907, 909–912, 914
\glslink .. 130 , 131, 1041, 1196, 1353	\glslongpluralaccesskey 466 , 665, 918
\glslink .. 130 , 131, 275, 283–285, 289, 460–463, 536, 605, 606, 612, 613, 616, 657–659, 787, 857, 858, 863–865, 869, 910, 912, 1041, 1112, 1118–1120, 1122, 1196, 1268, 1274–1276, 1278, 1353, 1433, 1439–1441, 1443	\glslongpluralkey 270 , 466, 603, 665, 855, 918, 1110, 1266, 1431
\glslinkcheckfirsthyperhook 131 , 131 , 537, 788, 1041, 1042, 1196, 1353	\glslongtok 274 , 275– 279, 283, 285, 456, 460, 462, 465, 605–608, 612, 613, 615– 623, 625–629, 653, 654, 657, 658, 661–664, 857–860, 863, 865, 867–876, 878–882, 906, 907, 910, 911, 914–918, 1112– 1115, 1118, 1119, 1267–1270, 1274, 1275, 1433–1436, 1439, 1440
\glslinkpostsetkeys 131 , 132, 537, 538, 788, 1042, 1196, 1197, 1353, 1354	\glslowercase 4 , 971, 1126, 1281
\glslinkvar 129 , 129 , 130, 536, 787, 1040, 1041, 1195, 1196, 1352, 1353	\glsLTpenaltycheck 320, 320 , 321, 697, 698, 951
\glslistdottedwidth 309 , 309 , 674, 692, 927, 945	\glsmakefirstuc . 45, 994, 1148, 1304
\glslistexpandedname .. 306 , 307 , 944	\glsmcols 329 , 329 , 330– 334, 701–704, 955–958, 1446–1449
\glslistgroupheaderfmt ... 306 , 306 , 307–309, 691, 692, 944, 945	\GlsMcolTreeSetup 327
\glslistinit 306 , 307, 944	\glsmeasuredepth 113 , 1032, 1187, 1344
\glslistnavigationitem 306 , 308, 309, 691, 944, 945	\glsmeasureheight 113 , 143, 1032, 1051, 1187, 1205, 1344, 1362
\glslocalreset 114 , 115, 524, 525, 775, 776, 1032, 1033, 1187, 1188, 1344, 1345	\glsmeasurewidth 113 , 366, 367, 369, 416– 418, 1032, 1187, 1344, 1453, 1454
\glslocalresetall 115 , 525, 776, 1033, 1188, 1345	\glsmfuaddmap 5 , 6 , 99, 120, 121, 131, 146, 148, 151–165, 167, 169, 171, 173, 258–260, 272, 273, 422, 423, 425, 426, 971, 972, 1022, 1023, 1036, 1041, 1052–1060, 1062–1064, 1066–1069, 1106,
\glslocalunset 114 , 116, 129, 525, 526, 546–	

1111, 1126, 1177, 1191, 1196, 1207–1218, 1220, 1261, 1262, 1266, 1267, 1281, 1334, 1348, 1353, 1364–1375, 1377, 1427, 1432	\glsnoexpandfields
\glsmfublocker 91 , 511, 761, 1017, 1172, 1329
. . . 5 , 6, 6 , 99, 147, 149, 151, 153–167, 169, 171, 173, 971, 972, 1023, 1052–1064, 1126, 1177, 1208–1216, 1218–1220, 1281, 1334, 1365–1373, 1375–1377	\glsnogroupskipfalse
\glsmfuexcl . . . 5 , 6, 176, 178–184, 186, 259, 260, 971, 972, 1071, 1126, 1221–1226, 1281, 1378–1383	. . . 15 , 472, 722, 976, 1131, 1285
\glsmoveentry	\glsnoidxdisplayloc
. 109 , 523, 773, 1030, 1184, 1341	208, 209, 250 , 579, 580, 595, 831, 832, 847, 1086, 1087, 1102, 1242, 1243, 1258, 1399, 1400, 1422
\GLSname 156 , 551, 802, 1056, 1212, 1369	\glsnoidxdisplaylocclishandler
\Glsname 155 , 155 , 551, 802, 1056, 1211, 1368	208, 250 , 579, 594, 831, 846, 1086, 1102, 1241, 1258, 1398, 1422
\glsname 155 , 155 , 551, 802, 1056, 1211, 1368	\GlsNoIdxDoRerunCheck
\glsnameaccessdisplay	208, 211
438 , 451, 454, 455, 638, 649, 650, 652, 653, 891, 903, 905, 906	\glsnoidxinithook
\glsnamefont 258 , 259, 265 , 454, 455, 598, 599, 601, 632, 652, 850, 851, 853, 885, 905, 1105, 1106, 1108, 1261, 1263, 1426, 1427, 1429 243 , 243
\glsnavhypergroupdotarget . 301 , 302	\glsnoidxitemhook
\glsnavhyperlink 300 , 303 , 688, 689, 941, 942, 1125, 1126 244 , 244
\glsnavhyperlinkname	\glsnoidxloclist 207 , 244 , 246 , 248 , 249 , 249 , 578, 594, 830, 845, 846, 1085, 1101, 1102, 1241, 1257, 1398, 1418, 1419, 1421, 1422
. . . 300 , 300 , 301 , 688, 941, 1125	\glsnoidxloclisthandler
\glsnavhypertarget 301 , 306, 308, 309, 330–334, 404, 412, 414, 415, 419, 688, 691, 692, 702– 705, 714, 715, 717, 941, 944, 945, 955–958, 967–969, 971, 1125, 1446–1449, 1451, 1452, 1454	249 , 594, 846, 1102, 1257, 1422
\glsnavigation 302 , 306, 308, 309, 330–334, 410, 412, 414, 415, 419, 688, 691, 702– 704, 714, 715, 717, 942, 944, 945, 955–958, 967–969, 971, 1125, 1446–1449, 1451, 1452, 1454	\GlsNoIdxMissingAction 243 , 243 , 247
\glsnavigationitem 303 , 303	\glsnoidxnumberlistloophandler
\glsnextpages 13 , 86 , 226, 255 , 472, 508, 587, 597, 721, 758, 839, 849, 976, 1014, 1095, 1104, 1130, 1169, 1250, 1260, 1285, 1326, 1407, 1425 208 , 209 , 579, 580, 831, 1086, 1087, 1242, 1399
	\glsnoidxprecontenthook
 243 , 244 , 1417
	\glsnoidxprenumberlist
 87 , 248, 249, 1014, 1101, 1169, 1257, 1326, 1421, 1422
	\glsnoidxstripaccents
	. . . 28 , 28 , 479, 480, 729, 730, 983, 984, 1138, 1139, 1293, 1294
	\glsnomakeindexwarning
	. 61 , 195, 495, 570, 745, 822, 1001, 1077, 1155, 1232, 1312, 1389
	\glsnonextpages
	86, 226, 255 , 508, 587, 597, 758, 839, 849, 1014, 1095, 1104, 1169, 1250, 1260, 1326, 1407, 1425
	\glsnopostdotfalse
	. . . 15 , 472, 722, 976, 1131, 1285
	\glsnoxindywarning
 56 , 61 , 63–65, 67–70, 188, 189, 492, 495–497, 499, 500, 565, 566, 742, 745–747, 749–751, 817, 818, 998, 1001– 1003, 1005, 1006, 1072, 1153, 1155–1158, 1160, 1161, 1227, 1228, 1310, 1312–1318, 1384, 1385

<code>\glsnumberformat</code>	56, 184, 492, 563, 564, 743, 815, 998, 1070, 1153, 1225, 1310, 1382	764, 817, 819, 821, 1019, 1020, 1071, 1074, 1075, 1174, 1227, 1230, 1231, 1331, 1384, 1387, 1388
<code>\glsnumberlistloop</code>	208, <u>209</u> , 579, 831, 1086, 1087, 1242, 1399	<code>\GLSpl</code> . <u>149</u> , 548, 800, 1053, 1209, 1366
<code>\glsnumbersgroupname</code>	45, <u>51</u> , 54, 263, 488, 490, 491, 600, 738, 740, 742, 852, 993, 996, 997, 1107, 1148, 1150, 1152, 1262, 1303, 1307, 1309, 1428	<code>\Glspl</code> 121, 148, <u>148</u> , 289, 529, 548, 615, 779, 799, 867, 1037, 1053, 1122, 1191, 1208, 1277, 1348, 1365, 1443
<code>\glsnumlistlastsep</code>	185, <u>185</u> , 207, 564, 579, 816, 830, 1070, 1086, 1226, 1241, 1383, 1398	<code>\glspl</code> 120, <u>147</u> , 148, 288, 528, 547, 615, 779, 799, 867, 1036, 1052, 1053, 1122, 1191, 1208, 1277, 1348, 1365, 1443
<code>\glsnumlistparser</code>	185, 204, 564, 577, 816, 828, 1070, 1084, 1226, 1239, 1383, 1396	<code>\GLSplural</code> <u>154</u> , 550, 802, 1055, 1211, 1368
<code>\glsnumlistsep</code>	185, <u>185</u> , 207, 564, 579, 816, 830, 1070, 1086, 1226, 1241, 1383, 1398	<code>\Glsplural</code> 153, <u>153</u> , 550, 802, 1055, 1211, 1368
<code>\glsopenbrace</code>	67, <u>187</u> , 192, 193, 499, 565, 568, 670, 671, 749, 816, 820, 923, 924, 1004, 1005, 1071, 1074, 1075, 1159, 1227, 1230, 1316, 1384, 1387	<code>\glsplural</code> 153, <u>153</u> , 550, 802, 1055, 1210, 1211, 1367, 1368
<code>\glsorder</code>	<u>35</u> , 36, 200–203, 240, 484, 574–576, 591, 733, 734, 825–828, 843, 988, 1080– 1083, 1098, 1143, 1236–1239, 1254, 1298, 1393, 1394, 1396, 1415	<code>\glspluralaccessdisplay</code> <u>439</u> , 442, 450, 638, 640, 641, 643, 644, 649, 891, 893, 894, 896, 897, 902
<code>\glsorg@endtheglossary</code> 6, 7, 468, 718, 972, 1127, 1281	<code>\glspluralsuffix</code> 51, <u>51</u> , 106, 107, 275, 456, 457, 461–464, 490, 521, 607, 608, 653, 654, 658, 660, 740, 771, 772, 859, 860, 907, 911– 913, 996, 1028, 1112, 1114, 1115, 1150, 1182, 1268, 1270, 1271, 1307, 1339, 1433, 1435, 1436
<code>\glsorg@PrintChanges</code> 6, 7, 468, 718, 972, 1127, 1281	<code>\glspostdescription</code> <u>14</u> , 54, 305, 307, 308, 311, 322, 335, 342, 343, 362, 386, 387, 411–415, 417, 418, 472, 492, 673–681, 690–693, 698, 705, 709, 713–717, 722, 742, 927, 929–935, 943–946, 952, 959, 963, 967–970, 976, 997, 1131, 1152, 1285, 1309, 1450–1454
<code>\glsorg@theglossary</code> 6, 7, 468, 718, 972, 1127, 1281	<code>\glspostinline</code> 304, <u>305</u> , 689, 690, 942, 943
<code>\glspagelistwidth</code>	<u>310</u> , 312, 313, 315, 316, 318–320, <u>321</u> , 323–326, <u>335</u> , 337, 338, 340, 341, <u>342</u> , 344–348, 692– 695, 697–701, 705–712, 946, 947, 949–955, 959, 960, 962–966	<code>\glspostlinkhook</code> . 130, <u>131</u> , 146– 151, 167–174, 447–449, 537, 546–549, 555–559, 647–649, 788, 798–801, 807–811, 900– 902, 1041, 1052–1054, 1060– 1065, 1196, 1207–1210, 1216– 1220, 1353, 1364–1367, 1373–1377
<code>\glspar</code> . 54, 492, 742, 997, 1152, 1309		
<code>\glspatchLToutput</code>	317–319, <u>320</u> , 696, 697, 949–951	
<code>\glspatchtabularx</code> <u>113</u> , 524, 775, 1032, 1187, 1344	
<code>\glspenaltygroupskip</code>	317, 318, <u>320</u> , 696, 697, 949–951	
<code>\glspercentchar</code> 94, <u>188</u> , 192, 193, 512, 513, 565, 568, 569, 763,		<code>\glsprefixsep</code> 424, <u>424</u> , 425–428, 685–688, 938–941

<code>\glsprestandardsort</code>	<code>\glsseesep</code>
..... 17, <u>17</u> , 473, 474,	222, <u>223</u> ,
723, 977, 978, 1132, 1286, 1287	585, 837, 1093, 1249, 1405, 1406
<code>\glsquote</code>	<code>\glsentencecase</code> <u>4</u> , 127, 131, 151,
48, <u>187</u> ,	175–177, 382–385, 971, 1040,
565, 817, 1071, 1227, 1305, 1384	1041, 1054, 1065, 1066, 1126,
<code>\glsrefentry</code>	1194–1196, 1209, 1221, 1222,
. <u>256</u> , 597, 849, 1105, 1260, 1425	1280, 1351–1353, 1366, 1378, 1379
<code>\glsreset</code>	<code>\glsSetAlphaCompositor</code>
<u>113</u> , 115, 524, 525, 775, 776,	... <u>56</u> , 492, 742, 998, 1153, 1310
1032, 1033, 1187, 1188, 1344, 1345	<code>\glsSetCompositor</code>
<code>\glsresetall</code> <u>55</u> , 492, 742, 998, 1153, 1309
. <u>115</u> , 525, 776, 1033, 1188, 1345	<code>\glssetexpandfield</code>
<code>\glsresetcurrcountfalse</code>	26,
..... 116, 1034, 1189, 1346	26, 30–32, 478, 480, 481, 728,
<code>\glsresetentrycounter</code>	730, 731, 982, 984, 985, 1137,
. <u>256</u> , 597, 849, 1104, 1260, 1425	1139, 1140, 1291, 1292, 1294, 1295
<code>\glsresetentrylist</code> 192, 243,	<code>\glssetnoexpandfield</code> 26, <u>26</u> ,
247, 255, <u>255</u> , 568, 593, 596,	27, 30, 31, 478, 480, 481, 728,
597, 670, 820, 844, 848, 849,	730, 731, 982, 984, 985, 1137,
923, 1074, 1100, 1104, 1230,	1139, 1140, 1291, 1292, 1294, 1295
1256, 1260, 1387, 1417, 1420, 1425	<code>\GlsSetQuote</code>
<code>\glsresetsubentrycounter</code>	103, 193, <u>195</u> , 518, 569, 570,
.... <u>256</u> , 257, 304, 597, 598,	573, 769, 820, 822, 825, 1025,
673, 690, 849, 850, 926, 943,	1075, 1077, 1079, 1180, 1231,
1104, 1105, 1260, 1261, 1425, 1426	1232, 1235, 1337, 1388, 1389, 1392
<code>\glsrestoreLOutput</code> ... <u>320</u> , 697, 951	<code>\glsSetSuffixF</code>
<code>\glssanitizeortfalse</code> <u>56</u> , 492, 742, 998, 1153, 1310
... 31, 481, 731, 985, 1140, 1295	<code>\glsSetSuffixFF</code>
<code>\glssanitizeorttrue</code> ... 31, 481,	... <u>56</u> , 492, 742, 998, 1153, 1310
730, 731, 985, 1139, 1140, 1295	<code>\glssettoctitle</code>
<code>\glsSavedGlossaryGroup</code> ... 264, <u>264</u>	.. <u>51</u> , 54, 225, 490, 491, 587,
<code>\glsavenumberlistfalse</code>	740, 741, 839, 995, 997, 1094,
... 13, 472, 721, 975, 1130, 1285	1150, 1152, 1250, 1307, 1308, 1407
<code>\glsavewritesfalse</code>	<code>\glssetwidest</code> ... <u>416</u> , 716, 969, 1453
... 42, 486, 736, 992, 1146, 1302	<code>\GlsSetWriteIstHook</code>
<code>\glssee</code> <u>222</u> , 585, 837, 1093, 1248, 1405	. <u>195</u> , 570, 822, 1077, 1232, 1389
<code>\glsseeformat</code> 191, 206, 208,	<code>\GlsSetXdyCodePage</code>
<u>222</u> , 567, 578, 579, 585, 670,	500, 750, 751, 1006, 1161, 1318
819, 829, 831, 837, 923, 1073,	<code>\GlsSetXdyFirstLetterAfterDigits</code>
1085, 1086, 1093, 1229, 1240, <u>188</u> ,
1242, 1248, 1386, 1397, 1399, 1405	565, 817, 1071, 1072, 1227, 1384
<code>\glsseeitem</code>	<code>\GlsSetXdyLanguage</code>
222, <u>223</u> , 585,	.. <u>69</u> , 500, 750, 1006, 1161, 1318
586, 837, 1093, 1249, 1405, 1406	<code>\GlsSetXdyLocationClassOrder</code> ..
<code>\glsseeitemformat</code> <u>68</u> , 499, 749, 1005, 1160, 1317
<u>223</u> , 586, 837, 1093, 1249, 1406	<code>\GlsSetXdyMinRangeLength</code> .. <u>189</u> ,
<code>\glsseelastsep</code>	566, 817, 818, 1072, 1228, 1385
222, <u>223</u> ,	<code>\GlsSetXdyNumberGroupOrder</code>
585, 837, 1093, 1249, 1405, 1406 <u>188</u> , 565, 566,
<code>\glsseelist</code>	817, 1072, 1227, 1228, 1384, 1385
<u>222</u> , 585, 837, 1093, 1248, 1405	

<code>\GlsSetXdyStyles</code>	69,	<code>\glssubentrycounterfalse</code>	
500, 750, 1006, 1160, 1161, 1317		... 16, 473, 723, 977, 1132, 1286	
<code>\glsshortaccessdisplay</code>		<code>\glssubentrycounterlabel</code>	
... 177, 440, 445–		... 257, 257, 597,	
448, 456–465, 560, 639, 645–		598, 849, 850, 1105, 1261, 1426	
648, 653–661, 812, 892, 899–		<code>\glssubentryitem</code>	
901, 906–914, 1066, 1222, 1379		... 257, 305, 307–309, 311,	
<code>\glsshortaccesskey</code>	465, 665, 918	312, 314, 322, 323, 325, 335,	
<code>\glsshortaccsupp</code> .	437, 438, 637, 890	337, 339, 343, 344, 346, 408,	
<code>\glsshortkey</code>	270, 465, 603,	412, 413, 415, 417, 598, 673–	
665, 855, 918, 1110, 1266, 1431		682, 690–694, 698–700, 705–	
<code>\glsshortplaccsupp</code>	437, 637, 890	707, 709–711, 713–716, 850,	
<code>\glsshortpluralaccessdisplay</code> . .		926–935, 943–948, 952–954,	
... 440, 444, 445, 447, 456,		959–961, 963–965, 967, 968,	
457, 461–463, 639, 645, 647,		970, 1105, 1261, 1426, 1450–1453	
653, 654, 657–659, 661, 892,		<code>\glssubgroupheading</code>	402
898, 900, 906, 907, 910–912, 914		<code>\GLSsymbol</code>	
<code>\glsshortpluralaccesskey</code> 159, 552, 804, 1057, 1213, 1370	
... 465, 665, 918		<code>\Glsymbol</code>	158,
<code>\glsshortpluralkey</code> 270, 465, 603,		158, 552, 803, 1057, 1213, 1370	
665, 855, 918, 1110, 1266, 1431		<code>\glssymbol</code>	158, 158, 552,
<code>\glsshorttok</code>	274, 275,	803, 1057, 1212, 1213, 1369, 1370	
465, 605, 615–623, 625–629,		<code>\glssymbolaccessdisplay</code> . .	439,
661–665, 857, 867–882, 914–		452, 455, 638, 641–643, 650,	
918, 1111, 1112, 1267, 1268, 1433		652, 653, 891, 895, 896, 903, 906	
<code>\glsshowaccsupp</code> 8, 9, 469, 470, 719,		<code>\glssymbolnav</code> . . .	303, 689, 942, 1126
720, 973, 974, 1128, 1282, 1283		<code>\GLSsymbolplural</code>	
<code>\glsshowtarget</code> . . .	8, 9, 469, 470,	. 160, 552, 804, 1058, 1213, 1370	
719, 973, 974, 1128, 1282, 1283		<code>\Glsymbolplural</code>	159,
<code>\glsshowtargetfont</code>	8,	159, 552, 804, 1057, 1213, 1370	
8, 469, 470, 719, 973, 1128, 1283		<code>\glssymbolplural</code>	159,
<code>\glsshowtargetfonttext</code>		159, 552, 804, 1057, 1213, 1370	
... 8, 8, 973, 1128, 1283		<code>\glssymbolpluralaccessdisplay</code> .	
<code>\glsshowtargetinner</code> 439, 452, 638,	
... 8, 8, 973, 1128, 1282, 1283		640, 641, 650, 891, 893, 894, 904	
<code>\glsshowtargetouter</code>	8,	<code>\glssymbolsgroupname</code>	
8, 470, 719, 973, 1128, 1282, 1283		44, 51, 54, 263, 487, 490, 491,	
<code>\glsshowtargetsymbols</code>		600, 738, 740, 742, 852, 993,	
. . 8, 8, 470, 719, 973, 1128, 1283		996, 997, 1107, 1147, 1150,	
<code>\glssortnumberfmt</code>		1152, 1262, 1303, 1307, 1309, 1428	
18, 18, 19, 474, 475, 723, 724,		<code>\glstarget</code>	
978, 979, 1132, 1133, 1287, 1288		. 112, 113, 258, 305–312, 314,	
<code>\glsspace</code>	271, 271,	322, 323, 325, 335, 337, 339,	
604, 855, 868, 1110, 1266, 1431		342–344, 346, 363, 389–393,	
<code>\glstepentry</code>		411–415, 417, 418, 598–600,	
256, 257, 597, 598, 849, 850,		673–682, 690–694, 698–700,	
1104, 1105, 1260, 1261, 1425, 1426		705–707, 709–711, 713–717,	
<code>\glstepsubentry</code>		850–852, 926–935, 943–948,	
256, 257, 597, 598, 849, 850,		952–954, 959–961, 963–965,	
1104, 1105, 1260, 1261, 1425, 1426			

967–970, 1031, 1032, 1105, 1186, 1261, 1343, 1426, 1450–1454	715, 717, 955–958, 966–969, 971, 1446–1449, 1451, 1452, 1454
\glstexpdfstring 175, <u>175</u> , 176, 180, 216, 1065, 1068, 1090, 1221, 1223, 1245, 1378, 1380, 1402	\glstreepredesc . . . 411, <u>411</u> , 413, 414, 713–715, 966–968, 1450–1452
\GLStext <u>151</u> , 549, 801, 1054, 1210, 1367	\GlsTreeSetup <u>348</u>
\Glstext <u>152</u> , <u>152</u> , 549, 801, 1054, 1210, 1367	\GlsTreeStarBox <u>349</u> , 350
\glstext <u>151</u> , <u>152</u> , 549, 801, 1054, 1210, 1367	\GlsTreeStarItemCounterBox 350, <u>350</u> , 351
\glstextaccessdisplay <u>438</u> , 443, 449, 638, 641, 642, 644, 645, 649, 891, 895, 897, 898, 902	\GlsTreeStarNameBox <u>350</u> , 351
\glstextformat <u>122</u> , 130, 132, 529, 537, 538, 780, 788, 789, 1037, 1041, 1042, 1192, 1196, 1197, 1349, 1353, 1354	\GlsTreeStarOuterBox <u>350</u> , 351
\glstextup 52, <u>270</u> , 463, 490, 603, 660, 740, 855, 913, 996, 1110, 1150, 1266, 1307, 1431	\GlsTreeStarSubItemCounterBox 350, 351, 352
\glstildechar . . . 63, <u>188</u> , 192, 193, 496, 565, 568, 746, 817, 819, 820, 1002, 1071, 1074, 1075, 1157, 1227, 1230, 1313, 1384, 1387	\GlsTreeStarSymbolBox <u>350</u> , 351
\glstranslatefalse 33, 482, 483, 732, 986, 987, 1141, 1142, 1296, 1297	\glstreesubitem <u>329</u> , <u>410</u> , 411, 701, 713, 955, 966, 1446, 1450
\glstranslatetrue 33, 34, 482, 483, 732, 733, 986, 987, 1141, 1142, 1296, 1297	\glstreesubsubitem 329, 411, 411, 701, 713, 955, 966, 1446, 1450
\glstreechildpredesc . . <u>411</u> , 412, 413, 713, 714, 966–968, 1450, 1451	\GlsTreeUpdateWidestNameAndSymbol <u>349</u>
\glstreegroupheaderfmt . . . 329– 334, <u>410</u> , 412–415, 419, 702– 705, 712–715, 717, 955–958, 966–969, 971, 1446–1452, 1454	\GlsTreeUpdateWidestNameOrSymbol <u>349</u>
\glstreeindent 413, <u>414</u> , 415, 417, 418, 677–679, 714– 717, 931, 932, 968, 970, 1451–1454	\glstype 131, 132, 146–150, 167–174, 447–449, 537, 546–549, 555– 559, 647, 648, 788, 798–800, 806–811, 900–902, 1041, 1042, 1052–1054, 1060–1065, 1196, 1197, 1207–1209, 1216–1220, 1353, 1354, 1364–1366, 1373–1377
\glstreeitem 329, 330, 410, <u>410</u> , 411, 701, 702, 712, 713, 955, 956, 966, 1446, 1447, 1450	\glsucmarkfalse 15, 472, 722, 976, 1131, 1286
\glstreenamebox <u>416</u> , 417, 418, 716, 717, 969, 970, 1453, 1454	\glsucmarktrue 15, 472, 722, 976, 1131, 1285
\glstreenamefmt 410, <u>410</u> , 411–414, 416– 418, 712–717, 966–970, 1449–1454	\glsunexpandedfieldvalue . . <u>101</u> , 307, 768, 944, 1024, 1179, 1336
\glstreenavigationfmt 330–334, <u>410</u> , 412, 414, 415, 419, 702–704, 712, 714,	\glsunset <u>114</u> , 116–118, 129, 524– 527, 546–549, 775–778, 798– 801, 1032, 1033, 1035, 1040, 1187–1190, 1195, 1344–1347, 1352
	\glsunsetall <u>116</u> , 526, 776, 1033, 1188, 1345
	\glsupacrpluralsuffix <u>52</u> , 279, 280, 286, 490, 609, 614, 618, 621, 623, 626, 740, 861, 866, 871, 874, 876, 879, 996, 1115, 1116, 1121, 1150, 1271, 1276, 1307, 1436, 1437, 1442
	\glsuppercase <u>5</u> , 59, 98, 123– 126, 152–166, 168, 170, 172, 174, 259, 260, 275, 283–285, 383–385, 427, 428, 442–446,

448–454, 459–461, 463, 971,
 999, 1022, 1037–1039, 1054–
 1063, 1065, 1112, 1118–1120,
 1126, 1154, 1177, 1192–1194,
 1210–1220, 1268, 1273, 1274,
 1276, 1281, 1311, 1334, 1349–
 1351, 1367–1377, 1433, 1439–1441
 \GlsUseAcrEntryDispStyle
 276, 277,
 279–282, 284, 286, 287, 457–
 459, 461, 463, 464, 606–612,
 614, 654, 655, 658–660, 858–
 862, 864, 866, 907–909, 911,
 913, 1113, 1115–1117, 1119–
 1121, 1269, 1271–1273, 1275–
 1277, 1434, 1436–1438, 1440–1442
 \GlsUseAcrStyleDefs
 276, 277, 279–282, 284,
 286, 287, 457–459, 461, 463–
 465, 606, 607, 609–612, 614,
 615, 654, 655, 658–660, 858–
 864, 866, 907–909, 911, 913,
 914, 1113, 1115–1117, 1119–
 1121, 1269, 1271–1273, 1275–
 1277, 1434, 1436–1438, 1440–1442
 \GLSuseri
 . 161, 553, 804, 1058, 1214, 1371
 \Glsuseri 160,
 160, 553, 804, 1058, 1213, 1370
 \glsuseri 160,
 160, 552, 804, 1058, 1213, 1370
 \glsuseriaccessdisplay
 440, 452, 639, 651, 892, 904
 \GLSuserii
 . 162, 553, 805, 1058, 1214, 1371
 \Glsuserii 161,
 161, 553, 805, 1058, 1214, 1371
 \glsuserii 161,
 161, 553, 804, 1058, 1214, 1371
 \glsuseriiaccessdisplay
 440, 453, 639, 651, 892, 904
 \GLSuseriii
 . 163, 553, 805, 1059, 1214, 1371
 \Glsuseriii 162,
 162, 553, 805, 1059, 1214, 1371
 \glsuseriii 162, 162,
 553, 805, 1058, 1059, 1214, 1371
 \glsuseriiiaccessdisplay
 440, 453, 639, 651, 892, 904
 \GLSuseriv
 . 164, 554, 805, 1059, 1215, 1372
 \Glsuseriv 163,
 163, 554, 805, 1059, 1215, 1372
 \glsuseriv 163, 163, 553,
 805, 1059, 1214, 1215, 1371, 1372
 \glsuserivaccessdisplay
 441, 453, 639, 651, 892, 904, 905
 \GLSuserv
 . 165, 554, 806, 1060, 1215, 1372
 \Glsuserv 164,
 164, 554, 806, 1059, 1215, 1372
 \glsuserv 164,
 164, 554, 805, 1059, 1215, 1372
 \glsuservaccessdisplay
 441, 454, 639, 651, 652, 893, 905
 \GLSuservi
 . 166, 554, 806, 1060, 1216, 1373
 \Glsuservi 165,
 165, 554, 806, 1060, 1215, 1372
 \glsuservi 165,
 165, 554, 806, 1060, 1215, 1372
 \glsuserviaccessdisplay
 441, 454, 639, 652, 893, 905
 \GlsWarnAddProtectedPageFmt ...
 215, 215, 1089, 1245, 1402
 \glswrallowprimitivemodsfalse .
 217, 1090, 1246, 1403
 \glswrallowprimitivemodstrue ..
 583, 835
 \glswrglossdisableanchorcmds ..
 216,
 220, 267, 1090, 1091, 1109,
 1245, 1247, 1264, 1402, 1404, 1430
 \glswrglossdisablelocationcmds
 209, 214, 214,
 1087, 1089, 1243, 1244, 1399, 1401
 \glswrglosslocationtarget
 216, 267,
 1090, 1109, 1245, 1264, 1402, 1430
 \glswrglosslocationtextfmt
 216, 267,
 1090, 1109, 1245, 1264, 1402, 1430
 \glswrite 189–195,
 203, 205, 212, 566–570, 576,
 580, 581, 668–672, 818–822,
 828, 832, 833, 922–925, 1072–
 1076, 1083, 1088, 1228–1232,
 1239, 1243, 1385–1389, 1395, 1400

<code>\glswritedefhook</code>	<code>\hyperm</code> 268 , 603 , 854 , 1109 , 1265 , 1430
. 94 , 102 , 513 , 518 , 763 , 769 ,	<code>\hypersc</code> 268 , 603 , 855 , 1109 , 1265 , 1430
1019 , 1025 , 1174 , 1180 , 1331 , 1337	<code>\hypersf</code> 268 , 603 , 855 , 1109 , 1265 , 1430
<code>\glswriteentry</code> 214 , 214 ,	<code>\hypersl</code> 268 , 603 , 855 , 1109 , 1265 , 1430
581 , 833 , 1088 , 1089 , 1244 , 1401	<code>\hypertarget</code>
<code>\glswritefiles</code> 42 ,	. 143 , 545 , 797 , 1051 , 1205 , 1362
211 , 486 , 580 , 736 , 832 , 991 ,	<code>\hypertt</code> 268 , 603 , 855 , 1109 , 1265 , 1430
992 , 1087 , 1146 , 1243 , 1302 , 1400	<code>\hyperup</code> 268 , 603 , 855 , 1109 , 1265 , 1430
<code>\glswriteglossarygroup</code> . 39 , 40 , 403	
<code>\glswriteglossarysubgroup</code> 39 , 40 , 403	
<code>\glsxindyfalse</code>	
. . . 36 , 484 , 734 , 988 , 1143 , 1298	
<code>\glsxindytrue</code>	
37 , 484 , 734 , 988 , 989 , 1143 , 1299	
<code>\glxtrgetgrouptitle</code> 402 , 403	
<code>\glxtrifemptyglossary</code> 1420	
<code>\group</code>	
231 , 234 , 403 , 404 , 410 , 1410 , 1412	
H	I
<code>\H</code> 29 , 480 , 729 , 984 , 1138 , 1293	<code>\IeC</code> 29 , 479 , 729 , 983 , 1138 , 1293
<code>\hangindent</code> 333 ,	<code>\if</code> 134 , 135 , 219 , 220 , 539 ,
334 , 403 , 406 , 408 , 410 , 413 –	584 , 668 , 790 , 836 , 921 , 1043 ,
415 , 417 – 419 , 677 – 679 , 704 ,	1044 , 1091 , 1198 , 1247 , 1355 , 1404
705 , 712 , 714 – 717 , 930 – 932 ,	<code>\if@endfor</code> 688 , 941 , 1125
958 , 966 – 968 , 970 , 971 , 1448 – 1454	<code>\if@endpe</code> 227 , 404 , 410 , 1251 , 1408
<code>\hbox</code> 113 , 309 , 524 , 674 , 692 ,	<code>\if@files</code> 37 , 989 , 1143 , 1299
775 , 927 , 945 , 1032 , 1187 , 1344	<code>\if@glsldebug</code>
<code>\hfill</code> 309 , 674 , 692 , 927 , 945 7 , 25 , 213 , 469 , 477 , 581 ,
<code>\hline</code> 311 – 313 , 315 , 322 – 324 ,	718 , 727 , 833 , 972 , 981 , 1088 ,
326 , 336 – 338 , 340 , 341 , 343 –	1127 , 1136 , 1244 , 1281 , 1291 , 1400
348 , 693 – 695 , 699 – 701 , 705 –	<code>\if@glsl@docloaded</code> 6 , 6 , 21 ,
712 , 946 – 948 , 952 – 955 , 959 – 966	468 , 475 , 487 , 718 , 725 , 737 ,
<code>\hsize</code> 309 , 310 , 321 ,	972 , 979 , 1127 , 1134 , 1281 , 1289
335 , 342 , 692 , 698 , 705 , 709 ,	<code>\if@glslisacronymlist</code>
945 , 946 , 952 , 958 , 959 , 962 , 963 23 , 131 , 477 , 537 , 726 , 788 ,
<code>\hspace</code> 410 , 411 , 713 , 966 , 1450	981 , 1041 , 1135 , 1196 , 1290 , 1353
<code>\hss</code> 309 , 674 , 692 , 927 , 945	<code>\if@openright</code>
<code>\ht</code> 320 , 697 , 951 61 , 494 , 744 , 1000 , 1155 , 1312
<code>\hyperbf</code> 268 , 603 , 855 , 1109 , 1265 , 1430	<code>\if@org@KV@glslink@local</code>
<code>\hyperdef</code> 50 , 489 , 739 , 995 , 1150 , 1307 128 , 129 , 1040 , 1195 , 1352
<code>\hyperemph</code>	<code>\ifbool</code> 42 , 73 ,
. 268 , 603 , 855 , 1109 , 1265 , 1430	476 , 483 , 486 , 502 , 529 – 531 ,
<code>hyperfirst</code> (option) 34	725 , 733 , 736 , 752 , 780 – 782 ,
<code>\hyperit</code> 268 , 603 , 855 , 1109 , 1265 , 1430	992 , 1007 , 1146 , 1162 , 1302 , 1319
<code>\hyperlink</code> 128 ,	<code>\ifboolexpr</code> 53 , 79 ,
144 , 267 , 536 , 545 , 602 , 787 ,	490 , 505 , 600 , 740 , 755 , 852 ,
797 , 854 , 1040 , 1051 , 1109 ,	996 , 1011 , 1151 , 1166 , 1308 , 1322
1195 , 1206 , 1264 , 1352 , 1363 , 1430	<code>\ifcase</code> 7 , 9 ,
<code>\hypermd</code> 268 , 603 , 855 , 1109 , 1265 , 1430	11 , 33 , 40 , 41 , 44 , 45 , 86 , 253 ,
	412 , 469 – 471 , 482 , 487 , 508 ,
	596 , 677 , 713 , 718 , 720 , 721 ,
	732 , 737 , 758 , 847 , 930 , 967 ,
	972 , 974 , 975 , 986 , 990 , 992 ,
	994 , 1014 , 1103 , 1127 , 1129 ,
	1141 , 1145 , 1147 , 1148 , 1169 ,
	1259 , 1281 , 1283 , 1284 , 1296 ,
	1301 , 1302 , 1304 , 1326 , 1424 , 1450
	<code>\ifcsdef</code> 32 , 54 , 61 , 89 ,
	90 , 97 – 102 , 112 , 213 , 237 , 241 ,

```

246, 276, 277, 437, 482, 491,
494, 510, 511, 515–518, 535,
581, 589, 591, 592, 601, 606,
637, 732, 741, 744, 760, 761,
765–769, 786, 833, 841, 843,
844, 853, 858, 890, 986, 997,
1000, 1016, 1017, 1021–1025,
1031, 1088, 1096, 1098–1100,
1113, 1125, 1141, 1152, 1155,
1171, 1176–1179, 1186, 1244,
1252, 1254, 1255, 1269, 1296,
1308, 1312, 1328, 1333–1336,
1343, 1401, 1413, 1416, 1420, 1434
\ifcsempy . . . . 75, 423, 503, 684,
753, 937, 938, 1009, 1163, 1320
\ifcsequal . . . . .
. . 75, 503, 753, 1009, 1163, 1320
\ifcsname . 101, 768, 1024, 1179, 1336
\ifcsstrequal . . . . .
. 102, 518, 769, 1025, 1179, 1336
\ifcsstring . . . . .
. 101, 517, 768, 1024, 1179, 1336
\ifcsundef . . . 10, 37, 50, 53, 57–
60, 72, 73, 81, 83, 85, 104, 107,
117, 128, 133, 134, 144, 211,
224, 237, 250, 251, 253, 257,
262, 264–266, 269, 276, 438–
441, 471, 474, 484, 489, 490,
492–494, 501, 502, 505–507,
519, 521, 526, 536, 538, 545,
546, 574, 575, 580, 586, 588,
589, 595, 598, 600, 601, 603,
606, 638–640, 672, 688, 720,
724, 734, 739, 741, 743, 744,
752, 756, 758, 770, 772, 777,
787, 789, 797, 826, 832, 838,
840, 841, 847, 850, 852, 853,
855, 858, 891–893, 926, 942,
975, 978, 988, 995, 996, 998–
1000, 1007, 1012, 1013, 1026,
1028, 1034, 1040, 1042, 1043,
1051, 1087, 1093, 1097, 1102,
1103, 1105–1109, 1113, 1125,
1129, 1133, 1143, 1150, 1151,
1153–1155, 1162, 1166–1168,
1181, 1183, 1189, 1195, 1197,
1198, 1206, 1207, 1243, 1249,
1252, 1258, 1261–1263, 1265,
1268, 1284, 1298, 1306–1308,
1310, 1312, 1319, 1323–1325,
1338, 1340, 1346, 1352, 1354,
1355, 1363, 1364, 1400, 1406,
1414, 1422–1424, 1426–1430, 1434
\ifcsvoid . . . . .
. 77, 262, 1009, 1164, 1321, 1427
\ifdef . . . . . 5, 47, 77, 78, 87,
93, 94, 113, 128, 175, 207, 208,
216, 270, 306, 410, 411, 424,
437, 504, 509, 512, 524, 536,
559, 561, 578, 579, 603, 637,
685, 712, 713, 754, 759, 763,
775, 787, 811, 813, 830, 831,
855, 890, 938, 944, 966, 971,
994, 1010, 1015, 1019, 1032,
1040, 1065, 1085, 1086, 1090,
1110, 1126, 1149, 1164, 1165,
1169, 1173, 1174, 1187, 1195,
1221, 1241, 1242, 1245, 1266,
1281, 1304, 1321, 1322, 1326,
1330, 1331, 1344, 1352, 1378,
1398, 1399, 1402, 1431, 1449, 1450
\ifdefempty . . . . .
24, 39, 60, 71, 76–78, 82, 83,
122, 124, 202, 203, 205, 247,
248, 274, 276, 282, 291, 441,
444, 459, 477, 485, 494, 501,
503, 504, 506, 529, 532, 533,
576, 577, 593, 594, 605, 606,
611, 616, 620, 622, 624, 640,
643, 645, 656, 727, 735, 744,
751, 753, 754, 756, 757, 780,
783, 784, 828, 829, 845, 846,
857, 858, 863, 869, 872, 874,
877, 893, 896, 898, 909, 981,
990, 1000, 1007, 1009, 1010,
1012, 1013, 1037, 1038, 1082–
1084, 1100, 1101, 1112, 1113,
1117, 1122, 1136, 1144, 1154,
1162, 1164, 1165, 1167, 1192,
1193, 1238, 1239, 1256, 1257,
1267, 1268, 1273, 1278, 1290,
1300, 1311, 1319, 1321, 1322,
1324, 1349, 1350, 1395, 1396,
1420, 1421, 1433, 1434, 1439, 1443
\ifdefequal . . . . . 38, 75–78,
89–91, 95, 105, 109, 248, 438,
484, 485, 502–504, 510, 511,
513, 520, 523, 594, 638, 734,
735, 753, 754, 760, 761, 764,
771, 774, 846, 891, 989, 1008–

```

1010, 1016, 1017, 1020, 1027, 1030, 1101, 1143, 1144, 1163– 1165, 1171, 1172, 1175, 1181, 1184, 1257, 1299, 1320–1322, 1328, 1329, 1332, 1338, 1341, 1421	\ifglsexindy@glslnumbers 70, 501, 751, 1006, 1161, 1318
\ifdefstrequal 101, 517, 768, 1024, 1179, 1336	\ifglssacrdescription 35, 627, 881, 988, 1142, 1298
\ifdefstring 14, 53, 79, 195, 200–202, 205, 239–241, 248–250, 429, 472, 490, 505, 574–576, 591–595, 632, 722, 740, 755, 825–827, 829, 842, 843, 845, 846, 885, 976, 996, 1011, 1080–1082, 1084, 1098– 1100, 1102, 1131, 1151, 1166, 1236–1238, 1240, 1254–1258, 1285, 1308, 1322, 1323, 1393, 1394, 1396, 1415, 1416, 1421, 1422	\ifglssacrdua 35, 618, 623, 626–628, 871, 876, 879, 881, 988, 1143, 1298
\ifdefvoid 27, 28, 37, 52, 108, 200, 201, 248, 249, 479, 522, 593, 594, 728, 729, 773, 845, 846, 983, 988, 996, 1029, 1080, 1081, 1101, 1137, 1138, 1143, 1151, 1184, 1236, 1237, 1256, 1257, 1292, 1293, 1298, 1307, 1341, 1393, 1394, 1421	\ifglssacrfootnote 35, 131, 537, 627, 628, 788, 881, 987, 1041, 1142, 1196, 1298, 1353
\ifdim 279, 320, 416, 608, 697, 716, 860, 951, 969, 1115, 1270, 1436, 1453	\ifglssacrfootnote 22, 475, 476, 725, 980, 1134, 1135, 1289
\iffalse 108, 115, 522, 525, 773, 776, 1029, 1033, 1184, 1188, 1341, 1345	\ifglssacrshortcuts 49, 289, 488, 615, 739, 867, 994, 1122, 1149, 1277, 1306, 1443
\IfFileExists 14, 32, 33, 201, 202, 227, 472, 482, 574, 575, 588, 722, 731, 732, 826, 827, 839, 976, 986, 1081–1083, 1095, 1130, 1140, 1141, 1237, 1238, 1251, 1285, 1296, 1394, 1395, 1408	\ifglssacrsmallcaps 35, 618, 619, 621, 623, 626, 870–872, 874, 876, 878, 879, 988, 1142, 1298
\ifglossaryexists 58, 69, 71, 74, 199–202, 493, 500–502, 573–575, 743, 750–752, 825– 827, 998, 1006–1008, 1080– 1082, 1153, 1161–1163, 1235– 1237, 1310, 1318–1320, 1392–1394	\ifglssacrsmaller 35, 618, 619, 622, 623, 870–872, 874, 876, 988, 1142, 1298
\ifgl@sanitize@description 30, 480, 730, 984, 1139, 1294	\ifglssautomake 40, 41, 47, 205, 485, 486, 577, 735, 736, 828, 990, 991, 994, 1084, 1145, 1146, 1149, 1239, 1300, 1301, 1304, 1396
\ifgl@sanitize@name 30, 480, 730, 984, 1139, 1294	\ifglssdescsuppressed 75, 304, 503, 689, 753, 943, 1009, 1163, 1320
\ifgl@sanitize@symbol 30, 480, 730, 985, 1139, 1294	\ifglssentrycounter 15, 16, 49, 256, 257, 473, 489, 597, 598, 722, 723, 739, 849, 850, 976, 977, 994, 1104, 1105, 1131, 1132, 1149, 1260, 1261, 1286, 1306, 1425, 1426
	\ifglssentryexists 73, 73, 74, 92, 93, 103, 105, 501, 502, 512, 518, 520, 752, 762, 763, 769, 771, 1007, 1008, 1018, 1019, 1025, 1027, 1162, 1163, 1172–1174, 1180, 1182, 1319, 1329–1331, 1337, 1339
	\ifglssesclocations . 135, 214–216, 582, 834, 1044, 1089, 1090, 1199, 1244, 1245, 1356, 1401, 1402
	\ifgl\$fieldcseq 102, 518, 769, 1025, 1179, 1336
	\ifgl\$fielddefeq 101, 517, 768, 1024, 1179, 1336
	\ifgl\$fieldeq 101, 517, 768, 1024, 1179, 1336
	\ifgl\$fieldvoid 77, 1009, 1164, 1321

\ifglshaschildren	952–954, 959–961, 963–965, 967–969, 971, 1450–1452, 1454
..... 74, 305, 502, 673, 690, 752, 926, 943, 1008, 1163, 1320	\ifglsnonumberlist
\ifglshasdesc 75, 304, 503, 690, 753, 943, 1009, 1163, 1320	. 254, 596, 848, 1104, 1259, 1424
\ifglshasfield	\ifglsnopostdot
.. 77, 504, 754, 1009, 1164, 1321	... 15, 472, 722, 976, 1131, 1285
\ifglshaslong 76, 119– 121, 176, 277–279, 282, 285, 456, 459, 462, 503, 528, 529, 560, 607, 608, 611, 613, 622, 653, 654, 656, 658, 753, 779, 780, 811, 859, 860, 863, 864, 875, 906, 907, 909, 911, 1009, 1036, 1037, 1066, 1113–1115, 1118, 1119, 1164, 1191, 1221, 1269, 1270, 1273, 1275, 1321, 1348, 1378, 1435, 1436, 1439, 1440	\ifglslsnumberline
\ifglshasparent ... 75, 230, 237, 244, 248, 416, 503, 589, 593, 715, 753, 841, 845, 969, 1009, 1097, 1101, 1163, 1252, 1256, 1320, 1409, 1414, 1418, 1421, 1453	.. 61, 495, 745, 1000, 1155, 1312
\ifglshasprefix 423, 424, 425, 427, 684–687, 937–940	\ifglslsresetcurrcount
\ifglshasprefixfirst 423, 424, 425, 427, 684–687, 938–940 116, 117, 1034, 1189, 1346
\ifglshasprefixfirstplural 423, 425, 426, 428, 684, 685, 687, 688, 938–941	\ifglslssanitizesort 27, 31, 478, 481, 728, 730, 731, 982, 985, 1137, 1139, 1140, 1292, 1295
\ifglshasprefixplural . 423, 425, 426, 428, 684–687, 937, 939–941	\ifglslssavenumberlist 89, 204, 223, 510, 577, 586, 760, 828, 837, 1016, 1083, 1093, 1170, 1239, 1249, 1327, 1396, 1406
\ifglshasshort . 76, 223, 503, 754, 1009, 1093, 1164, 1249, 1321, 1406	\ifglslssavewrites
\ifglshassymbol 76, 406, 408, 411–414, 417, 418, 503, 617, 620, 624, 713–717, 753, 869, 872, 877, 967, 968, 970, 1009, 1164, 1321, 1450–1454	. 42, 199, 213, 486, 573, 581, 736, 825, 833, 991, 1080, 1088, 1146, 1235, 1244, 1301, 1392, 1401
\ifglshyperfirst	\ifglslssubentrycounter 16, 49, 256, 257, 473, 489, 597, 598, 723, 739, 849, 850, 977, 994, 1104, 1105, 1132, 1149, 1260, 1261, 1286, 1306, 1425, 1426
. 131, 537, 788, 1041, 1196, 1353	\ifglstoc 61, 495, 745, 1000, 1155, 1312
\ifglsindexononlyfirst	\ifglstranslate
. 214, 581, 833, 1089, 1244, 1401	.. 32, 52, 482, 490, 731, 740, 986, 996, 1140, 1150, 1296, 1307
\ifglsnogroupskip	\ifglstocmark
.... 307, 311, 312, 314, 317, 318, 322, 324, 325, 335, 337, 339, 343, 345, 347, 412, 413, 415, 418, 691, 693, 694, 696, 699, 700, 705–707, 709–711, 713–715, 717, 944, 946–950,	59, 493, 743, 744, 999, 1154, 1311
	\ifglstocmark 73, 119, 122–124, 131, 187, 214, 424–428, 442– 444, 502, 528–533, 537, 565, 581, 617, 620, 622, 624, 640– 645, 685–687, 752, 778, 780– 784, 788, 816, 833, 869, 872, 875, 877, 893–898, 938–941, 1007, 1035, 1037, 1038, 1041, 1071, 1089, 1162, 1190, 1192, 1193, 1196, 1227, 1244, 1319, 1347, 1349, 1350, 1353, 1384, 1401
	\ifglswrallowprimitivemods 217, 218, 583, 835, 1090, 1246, 1403
	\ifglstocmark 16, 36, 55, 56, 62–65, 67–70, 82, 109, 110, 136, 188, 189, 195, 200, 201, 203, 204, 219, 221, 227, 484, 492, 495–501, 506, 523,

540, 565, 566, 570, 573, 574,	985, 995, 1000, 1042, 1083,
584, 585, 588, 667, 668, 734,	1125, 1140, 1149, 1155, 1197,
742, 745–751, 756, 774, 791,	1239, 1295, 1306, 1312, 1354, 1396
817, 818, 822, 825, 826, 835,	<code>\IfTrackedLanguage</code>
837, 840, 920–922, 977, 988,	. 195, 570, 822, 1077, 1232, 1389
998, 1001–1006, 1012, 1030,	<code>\IfTrackedLanguageFileExists</code> . .
1044, 1071, 1072, 1077, 1080, 53, 420, 491, 666,
1081, 1083, 1091, 1092, 1095,	683, 741, 919, 936, 997, 1151, 1308
1132, 1143, 1152, 1153, 1155–	<code>\iftrue</code> 108, 114, 522, 525, 773, 776,
1161, 1167, 1184, 1185, 1199,	1029, 1033, 1184, 1188, 1341, 1345
1227, 1228, 1232, 1236–1238,	<code>\ifundef</code> 15, 16, 80, 92, 104, 189, 193,
1247, 1248, 1251, 1286, 1298,	203, 473, 505, 512, 519, 566,
1309, 1310, 1312–1318, 1324,	569, 576, 722, 723, 755, 762,
1341, 1342, 1356, 1384, 1385,	770, 818, 821, 828, 976, 977,
1389, 1393–1395, 1403, 1405, 1408	1011, 1018, 1026, 1072, 1075,
<code>\ifignoredglossary</code> 83, 105,	1083, 1131, 1132, 1166, 1173,
108, 213, 224, 507, 519, 522,	1181, 1228, 1231, 1239, 1286,
581, 586, 757, 770, 773, 833,	1323, 1330, 1338, 1385, 1388, 1395
838, 1013, 1026, 1029, 1088,	<code>\IfValueTF</code> 58, 349
1094, 1167, 1181, 1184, 1244,	<code>\ifvmode</code> 186, 564, 816, 1071, 1226, 1383
1249, 1324, 1338, 1341, 1401, 1406	<code>\ifvoid</code> 320, 698, 951
<code>\ifin@</code> . . 46, 488, 738, 994, 1148, 1304	<code>\ifx</code> 15, 17, 23, 39,
<code>\ifinlistcs</code> 241,	45, 46, 50, 62, 64, 66, 67, 70,
251, 252, 592, 595, 844, 847,	71, 79, 105–108, 110–112, 133,
1099, 1102, 1255, 1258, 1416, 1423	134, 137–142, 177, 189, 192,
<code>\ifinner</code> . 8, 469, 719, 973, 1128, 1282	194, 196, 198, 212, 215, 216,
<code>\ifKV@glslink@hyper</code>	218–222, 225, 226, 264, 266,
. . . . 131–133, 537, 538, 788,	267, 291, 293, 428, 438, 473–
789, 1042, 1196, 1197, 1353, 1354	477, 488, 489, 495, 496, 498,
<code>\ifKV@glslink@local</code> 128, 129, 546–	500, 501, 519–523, 538, 540–
549, 798–800, 1040, 1195, 1352	545, 560, 566, 568, 570–572,
<code>\ifmeasuring@</code>	580, 582–585, 587, 600–602,
. 111, 524, 775, 1031, 1185, 1342	616, 618, 619, 621, 623–625,
<code>\ifmmode</code> 8,	627, 629, 631, 637, 668–673,
469, 719, 973, 1128, 1282, 1283	677–679, 722–724, 726, 735,
<code>\ifnum</code> 117, 118, 202, 247, 320, 413,	738, 739, 745, 746, 748, 749,
415, 417, 474, 485, 527, 576,	751, 770–774, 789, 791–797,
593, 677–679, 697, 714–716,	811, 812, 818, 820, 822–824,
723, 724, 736, 777, 778, 827,	832, 834–837, 839, 852–854,
845, 931, 932, 951, 968, 970,	868, 870, 872, 874, 876–878,
978, 1034, 1035, 1082, 1100,	880, 882, 885, 891, 922, 924–
1132, 1189, 1190, 1238, 1256,	926, 930–932, 976–981, 990,
1346, 1347, 1395, 1421, 1451–1454	993–995, 1001, 1002, 1004,
<code>\ifstrempty</code> 9, 86, 470, 673,	1006, 1007, 1011, 1026–1031,
720, 926, 974, 1128, 1283, 1325	1042, 1043, 1045–1050, 1066,
<code>\ifstrequal</code> 600, 852	1072, 1075–1077, 1079, 1087,
<code>\ifthenelse</code> 32, 50, 60,	1089–1095, 1107–1109, 1122,
133, 204, 481, 489, 494, 538,	1131–1133, 1135, 1144, 1148,
576, 600, 628, 688, 731, 739,	1150, 1156–1159, 1161, 1162,
744, 789, 828, 852, 881, 941,	1165, 1181–1183, 1185, 1186,

1197, 1198, 1200–1205, 1221, 1222, 1228, 1230, 1232–1235, 1243, 1245–1248, 1250, 1263, 1264, 1278, 1286, 1287, 1290, 1300, 1303, 1304, 1307, 1313– 1316, 1318, 1319, 1322, 1338– 1340, 1342, 1343, 1354, 1355, 1357–1362, 1378, 1379, 1385, 1387, 1389–1392, 1400, 1402– 1405, 1407, 1428–1430, 1443, 1444	1152, 1228, 1231, 1237–1239, 1309, 1385, 1388, 1393, 1395, 1396
<code>\immediate</code>	<code>\item</code> 306–310, 329, 330, 411, 412, 673, 674, 676, 677, 691, 692, 701, 702, 713, 714, 926, 927, 930, 944–946, 955, 956, 966, 967, 1446, 1447, 1450, 1451
39, 92, 94, 119, 200, 212, 228, 485, 512, 513, 528, 573, 575, 581, 588, 735, 762, 763, 778, 779, 825, 827, 832, 833, 840, 990, 1018, 1019, 1035, 1036, 1080–1083, 1088, 1095, 1096, 1144, 1145, 1173, 1174, 1190, 1236–1238, 1243, 1251, 1300, 1330, 1331, 1347, 1393, 1400, 1408	J <code>\jobname</code> 39, 55, 92–94, 189, 193, 200– 203, 227, 485, 492, 512, 566, 569, 573–575, 588, 668, 671, 735, 742, 762, 763, 818, 821, 825–827, 839, 922, 924, 990, 998, 1018, 1019, 1072, 1075, 1080–1083, 1095, 1144, 1152, 1173, 1174, 1228, 1231, 1236– 1238, 1251, 1300, 1309, 1330, 1331, 1385, 1388, 1393–1395, 1408
<code>\in@</code> 46, 488, 738, 994, 1148, 1304	K
<code>index</code> (option) 45	<code>kernelglossredefs</code> (option) 43
<code>index</code> (style) 411	<code>\key@ifundefined</code> 96, 97, 514, 764, 765, 1020, 1021, 1175, 1176, 1332, 1333
<code>indexgroup</code> (style) 412	<code>\keys</code> 12, 327, 348, 378
<code>indexhypergroup</code> (style) 412	keys:
<code>\indexname</code> 45, 488, 738, 993, 1148, 1303	<code>access</code> 430
<code>indexonlyfirst</code> (option) 34	<code>counter</code> 85
<code>\indexspace</code> 306, 307, 329, 329, 330–334, 410 , 412–415, 418, 419, 690, 691, 701–705, 712–715, 717, 944, 955–958, 966–969, 971, 1446–1452, 1454	<code>description</code> 83
<code>inline</code> (style) 303	<code>descriptionaccess</code> 430
<code>\input</code> . 52, 122, 490, 529, 740, 780, 996, 1037, 1150, 1192, 1307, 1349	<code>descriptionplural</code> 84
<code>\inputencodingname</code> 37, 484, 734, 988, 1143, 1298	<code>descriptionpluralaccess</code> 431
<code>\InputIfFileExists</code> 93, 512, 763, 1019, 1173, 1330	<code>first</code> 84
<code>\int</code> 18, 19, 48, 230, 244, 245, 251, 252, 263, 327, 352–358, 364, 365, 402–406, 408, 1287, 1288, 1305, 1418, 1423	<code>firstaccess</code> 430
<code>\istfile</code> 211 , 580, 832	<code>firstplural</code> 84
<code>\istfilename</code> 55, 55, 189, 193, 201–203, 492, 566, 569, 574–576, 668, 671, 742, 818, 821, 826–828, 922, 924, 998, 1072, 1075, 1081–1083,	<code>firstpluralaccess</code> 430
	<code>long</code> 88
	<code>longaccess</code> 431
	<code>longplural</code> 88
	<code>longpluralaccess</code> 431
	<code>name</code> 83
	<code>nonumberlist</code> 86
	<code>parent</code> 86
	<code>plural</code> 84
	<code>pluralaccess</code> 430
	<code>see</code> 85
	<code>short</code> 88
	<code>shortaccess</code> 431

shortplural	88	\leavevmode	102, 132, 518, 537, 769, 788,
shortpluralaccess	431		1025, 1042, 1180, 1197, 1337, 1354
sort	84	\legacy	48, 402, 405, 1305
symbol	85	\let	6, 7, 11, 14, 17–
symbolaccess	430		22, 28–30, 32–34, 37–39, 41–
symbolplural	85		46, 49, 50, 53, 54, 64, 65, 77–
symbolpluralaccess	430		79, 89, 91, 102–108, 111–116,
text	84		118, 122, 128–130, 132, 133,
textaccess	430		135, 143–151, 166–174, 177,
type	85		184, 185, 193, 195, 199–204,
user1	87		206–208, 213–216, 218, 222,
user1access	431		224–227, 249, 253, 255, 264,
user2	87		274, 287–289, 302, 303, 306,
user2access	431		320, 329, 330, 410, 411, 419–
user3	87		422, 432, 433, 447–449, 468,
user3access	431		469, 472, 474–476, 479, 480,
user4	88		482, 483, 485–490, 492, 496,
user4access	431		497, 504, 505, 510, 511, 518–
user5	88		522, 524–527, 529, 536–540,
user5access	432		545–549, 555–560, 563, 564,
user6	88		569, 570, 573–579, 581–583,
user6access	432		585–587, 594, 595, 597, 599,
\KV@glslink@hyperfalse	128,		605, 615–627, 634, 647, 648,
	131, 144, 536, 537, 546, 787,		661–666, 672, 683, 684, 688,
	788, 797, 1040–1042, 1051,		689, 697, 701, 702, 713, 718,
	1195, 1196, 1207, 1352, 1353, 1364		719, 722–725, 729–733, 735–
\KV@glslink@hypertrue			742, 747, 754, 755, 760–762,
	128, 145, 536, 546, 787, 797,		769–773, 775–778, 780, 787–
	1040, 1051, 1195, 1207, 1352, 1364		791, 797–801, 806–812, 815,
L			
\L	29, 480, 729, 984, 1138, 1294		816, 820, 822, 825–831, 833–
\l	5, 12, 27, 29, 31,		835, 837–839, 846, 847, 849,
	47, 48, 112, 228–235, 242–246,		851, 857, 867, 868, 870–878,
	251, 327–329, 352–361, 363–		880, 887, 888, 900–902, 914–
	366, 368–382, 389–393, 396–		919, 925, 936, 937, 941, 942,
	410, 480, 729, 972, 984, 1126,		944, 951, 955, 956, 966, 972,
	1138, 1186, 1281, 1294, 1305,		973, 976–980, 983, 984, 986–
	1343, 1408–1412, 1417–1419, 1423		997, 1002, 1003, 1010, 1011,
\label	11, 111–		1016–1018, 1025–1029, 1031–
	113, 253, 254, 256, 471, 596,		1035, 1037, 1040–1042, 1044,
	597, 721, 848, 849, 975, 1031,		1051–1054, 1060–1066, 1070,
	1032, 1103, 1104, 1130, 1186,		1075, 1076, 1080–1090, 1093–
	1259, 1260, 1284, 1343, 1424, 1425		1095, 1101–1104, 1112, 1121,
\language			1122, 1125–1127, 1131–1134,
\language (option)	34		1138–1152, 1157, 1158, 1165,
\leaders	309, 674, 692, 927, 945		1171, 1172, 1180–1184, 1186–
			1192, 1195–1197, 1199, 1206–
			1210, 1216–1220, 1222, 1225,
			1226, 1231, 1232, 1235–1242,
			1244–1246, 1248–1251, 1257,

1258, 1260, 1267, 1277, 1281, 1282, 1285, 1287–1289, 1293, 1294, 1296–1304, 1306, 1308, 1309, 1314, 1315, 1321, 1322, 1328, 1329, 1337–1341, 1343– 1349, 1352–1354, 1356, 1363– 1367, 1373–1377, 1379, 1382, 1383, 1388, 1389, 1392–1394, 1396–1399, 1401–1403, 1405– 1408, 1421, 1423, 1425, 1433, 1442, 1443, 1446, 1447, 1450	<code>\letcs</code> 37, 39, 75–78, 94, 96, 97, 100, 106, 107, 174–176, 207, 208, 237–239, 245, 248, 258, 262, 416, 502–504, 512, 514, 517, 521, 559, 560, 574, 575, 578, 579, 589, 590, 593, 594, 598, 600, 716, 735, 753, 754, 763, 765, 768, 771, 772, 811, 826, 827, 830, 831, 841, 842, 845, 846, 850, 852, 969, 988, 990, 1008–1010, 1019, 1021, 1024, 1027, 1028, 1065, 1066, 1085, 1086, 1096–1098, 1101, 1105– 1107, 1125, 1143, 1145, 1163– 1165, 1174–1176, 1179, 1182, 1183, 1220, 1221, 1241, 1242, 1252, 1253, 1256, 1257, 1261, 1262, 1298, 1300, 1320–1322, 1331–1333, 1336, 1339, 1340, 1377, 1378, 1398, 1399, 1413– 1415, 1419, 1421, 1426, 1427, 1453	<code>\long</code> . 102, 268, 518, 603, 769, 854, 1025, 1109, 1180, 1265, 1337, 1430
	<code>long</code> (key) 88	
	<code>long</code> (style) 310	
	<code>long-booktabs</code> (style) 317	
	<code>long-sc-short</code> (acrstyle) 279	
	<code>long-sc-short-desc</code> (acrstyle) 281, 457	
	<code>long-short</code> (acrstyle) 277, 456	
	<code>long-short-desc</code> (acrstyle) . . 280, 457	
	<code>long-sm-short</code> (acrstyle) 279	
	<code>long-sm-short-desc</code> (acrstyle) 281, 458	
	<code>long-sp-short</code> (acrstyle) 278	
	<code>long-sp-short-desc</code> (acrstyle) . . 280	
	<code>long3col</code> (style) 312	
	<code>long3col-booktabs</code> (style) 317	
	<code>long3colborder</code> (style) 313	
	<code>long3colheader</code> (style) 313	
	<code>long3colheaderborder</code> (style) 313	
	<code>long4col</code> (style) 313	
	<code>long4col-booktabs</code> (style) 318	
	<code>long4colborder</code> (style) 314	
	<code>long4colheader</code> (style) 314	
	<code>long4colheaderborder</code> (style) 315	
	<code>longaccess</code> (key) 431	
	<code>longborder</code> (style) 311	
	<code>longheader</code> (style) 311	
	<code>longheaderborder</code> (style) 311	
	<code>\longnewglossaryentry</code> 102, 103, 518, 769, 1025, 1180, 1337	
	<code>longplural</code> (key) 88	
	<code>longpluralaccess</code> (key) 431	
	<code>\longprovideglossaryentry</code> 103, 518, 769, 1025, 1180, 1337	
	<code>longragged</code> (style) 321	
	<code>longragged-booktabs</code> (style) 319	
	<code>longragged3col</code> (style) 323	
	<code>longragged3col-booktabs</code> (style) . 319	
	<code>longragged3colborder</code> (style) . . . 324	
	<code>longragged3colheader</code> (style) . . . 324	
	<code>longragged3colheaderborder</code> (style) 324	
	<code>longraggedborder</code> (style) 322	
	<code>longraggedheader</code> (style) 322	
	<code>longraggedheaderborder</code> (style) . . 323	
	<code>\LT@end@open</code> 320, 697, 951	
	<code>\LT@err</code> 320, 697, 951	
	<code>\LT@foot</code> 320, 697, 698, 951	
	<code>\LT@head</code> 320, 321, 697, 698, 951	
	<code>\LT@lastfoot</code> 320, 697, 698, 951	
	<code>\LT@output</code> 320, 697, 951	

M

```

\makeatletter .....
. 93, 227, 512, 588, 763, 839,
1019, 1095, 1173, 1251, 1330, 1408
\makebox .. 309, 351, 352, 416–418,
674, 678, 679, 692, 716, 717,
927, 932, 945, 969, 970, 1453, 1454
\makefirstuc ..... 4, 971, 1126, 1280
\makeglossaries ..... 9, 10, 38,
40, 44, 50, 86, 199, 203, 205,
207, 209, 228, 470, 484–487,
490, 508, 573, 576–578, 580,
588, 720, 734–737, 740, 758,
825, 828–830, 832, 840, 974,
989, 990, 992, 993, 995, 1014,
1080, 1083–1085, 1087, 1096,
1129, 1143–1145, 1147, 1150,
1169, 1236, 1238–1242, 1251,
1283, 1299, 1300, 1302, 1303,
1307, 1326, 1393, 1395–1399, 1408
\makeglossary .....
... 43, 43, 44, 486, 487, 736,
737, 992, 993, 1147, 1302, 1303
makeindex (option) ..... 36
\makenoidxglossaries ..... 9,
10, 38, 86, 204, 205, 209, 255,
470, 484, 485, 508, 577–580,
720, 734, 735, 758, 828–832,
974, 989, 1014, 1084–1087,
1129, 1143, 1144, 1169, 1239–
1242, 1283, 1299, 1326, 1396–1399
\MakeTextUppercase .....
.... 5, 468, 718, 972, 1126, 1281
\MakeUppercase .....
641, 642, 648, 894–896, 901, 902
\marginpar 8, 470, 719, 973, 1128, 1283
\markboth 59, 493, 743, 999, 1154, 1311
\mbox ..... 186, 308, 333, 334,
416, 564, 674, 691, 704, 716,
816, 927, 945, 957, 958, 970,
1071, 1226, 1383, 1448, 1449, 1453
mcolalttree (style) ..... 333
mcolalttreegroup (style) ..... 333
mcolalttreehypergroup (style) ... 333
mcolalttreespannav (style) ..... 334
mcolindex (style) ..... 329
mcolindexgroup (style) ..... 329
mcolindexhypergroup (style) ..... 329
mcolindexspannav (style) ..... 330
mcoltree (style) ..... 330

```

```

mcoltree* (style) ..... 327
mcoltreegroup (style) ..... 331
mcoltreehypergroup (style) ..... 331
mcoltreename (style) ..... 331
mcoltreenamegroup (style) ..... 332
mcoltreenamehypergroup (style) ..... 332
mcoltreenamehypergroup (style) ..... 332
mcoltreenamespannav (style) ... 332
mcoltreespannav (style) ..... 331
\memUchead 59, 493, 743, 999, 1154, 1311
mendex (option) ..... 36
\MessageBreak ..... 25, 43, 80,
    225, 255, 420, 421, 428, 429,
    478, 486, 487, 505, 587, 631,
    666, 683, 727, 736, 737, 755,
    839, 885, 919, 920, 936, 982,
    992, 1011, 1094, 1136, 1147,
    1166, 1250, 1291, 1302, 1323, 1407
mffirstuc (option) ..... 45
\mffirstucMakeUppercase ..... 5,
    468, 493, 516, 530–534, 549–
    559, 606, 611–613, 620, 624,
    644–646, 649–652, 656, 657,
    659, 687, 688, 718, 743, 744,
    766, 781–785, 801–809, 811,
    857, 858, 863–865, 873, 877,
    897–899, 902–905, 909, 910,
    912, 940, 941, 972, 1126, 1281
\MFUaddmap ..... 5, 971, 1126, 1281
\MFUblocker ..... 5, 971, 1126, 1281
\MFUexcl ..... 5, 971, 1126, 1281
\MFUsentencecase ..... 5,
    5, 45, 175, 176, 180, 971, 972,
    994, 1065, 1068, 1126, 1148,
    1221, 1223, 1281, 1304, 1378, 1380
\midrule ..... 317, 318, 696, 949, 950
\month ..... 189, 193, 566, 569,
    668, 671, 818, 821, 922, 924,
    1072, 1075, 1228, 1231, 1385, 1388
/msg ..... 229, 1409

```

N

\n 194, 569, 671, 672, 821,
 924, 925, 1075, 1076, 1231, 1388
 name (key) 83
 \NeedsTeXFormat . 4, 299, 303, 306,
 310, 316, 321, 327, 334, 341,
 348, 419–421, 428, 467, 468,
 631, 666, 667, 672, 683, 718,
 884, 919, 920, 926, 936, 944,
 971, 1125, 1126, 1280, 1446, 1449

<code>\new@glossaryentry</code>	91, 92 , 207, 511, 512, 578, 762, 830, 1018, 1085, 1172, 1173, 1241, 1329, 1330, 1398	825–829, 831–860, 867–885, 889–891, 906, 914, 918, 919, 926, 937–944, 951, 955, 966, 969, 971–1022, 1024, 1025, 1029–1038, 1040–1042, 1044, 1051–1072, 1077, 1079–1084, 1087–1113, 1115, 1121–1126, 1128–1150, 1152–1177, 1179, 1180, 1184–1197, 1199, 1205– 1228, 1232, 1235–1238, 1240, 1242–1263, 1265–1270, 1277– 1298, 1300–1304, 1306, 1307, 1309–1334, 1336, 1337, 1341– 1354, 1356, 1362–1385, 1389, 1392–1397, 1399–1408, 1413– 1417, 1420–1422, 1424–1434, 1436, 1442–1446, 1449, 1450, 1453
<code>\new@ifnextchar</code>	. 80, 97, 98, 119– 121, 145–149, 151–173, 272, 273, 424–427, 505, 515, 516, 528, 546–559, 603, 604, 685– 687, 755, 766, 779, 798–810, 855, 856, 938–940, 1011, 1022, 1036, 1051–1064, 1110, 1111, 1166, 1176, 1177, 1191, 1207– 1220, 1266, 1267, 1323, 1333, 1334, 1348, 1364–1377, 1431, 1432	
<code>\newacronym</code> 269, 269 , 270, 274, 603, 605, 616, 618, 619, 621, 623, 625, 627, 629, 855, 857, 868, 870, 872, 874, 876, 878, 880, 882, 1109, 1110, 1112, 1265, 1267, 1430, 1431, 1433	
<code>\newacronymhook</code>	.. 274 , 275, 465, 605, 616, 618, 620, 621, 623, 626, 627, 629, 661, 857, 868, 870, 872, 874, 876, 879, 880, 882, 914, 1112, 1267, 1268, 1433	
<code>\newacronymstyle</code> 276 , 277–282, 284, 286, 287, 606– 612, 614, 858–864, 866, 1113– 1117, 1119–1121, 1269–1273, 1275–1277, 1434–1438, 1440–1442	
<code>\newcommand</code> 4–6, 8–32, 34–41, 43–47, 49–52, 54–65, 67–75, 77–84, 86–98, 101–103, 108– 122, 124, 126–131, 133, 135, 142–189, 195, 198–203, 205, 209–216, 218–225, 227, 236– 244, 246, 248–250, 254–265, 268–274, 276–278, 287, 289– 291, 293–303, 305–307, 320, 329, 349, 350, 410, 411, 416, 422–427, 429, 434–438, 455, 465–467, 469–479, 481–503, 505–518, 522–529, 532, 533, 535–540, 545–566, 570, 573– 577, 579–586, 588–601, 603– 608, 615–632, 636–638, 653, 661, 665, 666, 672, 684–691, 697, 701, 712, 713, 716, 719– 729, 731–733, 735–740, 742– 747, 749–753, 755–769, 773– 780, 783–791, 797–817, 822,	
<code>\newcount</code> 89, 474, 510, 724, 760, 978, 1016, 1133, 1171, 1328	
<code>\newcounter</code> 15, 16, 473, 722, 723, 976, 977, 1131, 1132, 1286	
<code>\NewDocumentCommand</code> 58, 143, 211, 327, 348, 349, 416, 1206, 1363	
<code>\newenvironment</code> 257, 598, 850, 1105, 1261, 1426	
<code>\newglossary</code> 21, 22, 44, 45, 80 , 81, 82, 204, 475, 476, 487, 488, 505, 506, 576, 725, 726, 738, 755, 756, 828, 979, 980, 993, 1011, 1012, 1083, 1134, 1135, 1147, 1148, 1166, 1167, 1239, 1289, 1303, 1323, 1324, 1396	
<code>\newglossaryentry</code>	9, 45, 88, 91, 91 , 117, 275, 470, 488, 509, 511, 526, 605, 615, 617, 618, 620, 622, 624, 626, 628, 661–664, 720, 738, 759–762, 777, 857, 867, 869, 871, 873, 875, 877, 879, 881, 914–917, 974, 993, 1015–1018, 1034, 1112, 1129, 1148, 1170, 1172, 1189, 1268, 1283, 1303, 1327, 1329, 1346, 1433	
<code>\newglossarystyle</code> 265 , 303, 307–319, 321– 326, 328–347, 402, 411–416, 418, 419, 601, 689, 691–717, 853, 942, 944–950, 952–969, 971, 1107, 1263, 1428, 1446–1454	
<code>\newif</code>	6, 7, 23, 32, 36, 40, 116, 217, 468, 469, 477, 482, 484, 485,	

583, 718, 726, 731, 734, 735, 835, 972, 981, 986, 988, 990, 1034, 1090, 1127, 1135, 1140, 1143, 1145, 1189, 1246, 1281, 1290, 1296, 1298, 1300, 1346, 1403					
<code>\newignoredglossary</code>				<code>\noalign</code>	320, 697, 951
.. 82, 506, 756, 1012, 1167, 1324				<code>\nobreak</code>	308,
<code>\newlabel</code>	143, 1206, 1363			321, 674, 691, 698, 927, 945, 951	
<code>\newlength</code>	142, 309,			<code>\noexpand</code> 23, 50, 62, 64, 93, 107–109,	
310, 321, 335, 342, 414, 545,				112, 133–136, 142, 184, 196–	
692, 698, 705, 709, 714, 797,				198, 204, 215, 220, 223, 227,	
945, 946, 952, 958, 959, 962,				228, 238, 239, 267, 269, 275,	
963, 968, 1051, 1205, 1362, 1452				476, 489, 495, 496, 512, 521,	
<code>\newrobustcmd</code>	9, 75–77, 91,			522, 535, 536, 538–540, 545,	
92, 98–101, 119–121, 130, 145–				564, 570–575, 577, 582, 583,	
173, 175, 176, 180, 186, 187,				586, 588, 590, 598, 599, 603,	
222, 223, 262, 266, 270–273,				605, 615, 617–622, 624–626,	
291, 415, 424–427, 438–441,				628, 632, 661–665, 667, 726,	
470, 502–504, 511, 512, 515,				740, 746, 763, 772, 773, 786,	
516, 528, 536, 546–565, 585,				789–791, 796, 797, 815, 822–	
586, 603, 604, 638–640, 684–				828, 834, 835, 838, 840, 842,	
687, 715, 719, 720, 752–754,				850, 851, 855, 857, 867, 869–	
761, 762, 766, 779, 787, 798–				871, 873, 875, 877–879, 881,	
816, 837, 855, 856, 891–893,				882, 885, 914–918, 920, 980,	
937–940, 969, 974, 1008, 1009,				995, 1001, 1002, 1019, 1028,	
1017, 1018, 1022–1024, 1036,				1029, 1031, 1042–1045, 1050,	
1041, 1051–1066, 1068, 1071,				1070, 1077–1084, 1089, 1092,	
1093, 1106, 1108, 1110, 1111,				1093, 1095–1098, 1109, 1112,	
1122, 1128, 1163, 1164, 1172,				1135, 1150, 1156, 1157, 1173,	
1173, 1176–1179, 1190, 1191,				1183, 1184, 1186, 1197–1200,	
1196, 1207–1221, 1223, 1226,				1205, 1226, 1232–1239, 1245,	
1227, 1248, 1249, 1262, 1263,				1247, 1249, 1251, 1253, 1264,	
1266, 1267, 1278, 1283, 1320,				1265, 1268, 1290, 1307, 1313,	
1321, 1329, 1330, 1333–1336,				1314, 1330, 1340, 1341, 1343,	
1347, 1348, 1353, 1364–1378,				1354–1357, 1362, 1383, 1389–	
1380, 1383, 1384, 1405, 1406,				1392, 1396, 1402, 1404, 1406,	
1427, 1429, 1431, 1432, 1443, 1452				1408, 1414, 1415, 1430, 1431, 1433	
<code>\newterm</code> 45, 488, 738, 993, 1148, 1303				<code>noglossaryindex</code> (option)	<u>45</u>
<code>\newtoks</code>	136, 199, 274,			<code>nogroupskip</code> (option)	<u>15</u>
540, 573, 605, 791, 825, 857,				<code>\nohyperpage</code>	266, 602,
1045, 1080, 1111, 1112, 1200,				853, 854, 1108, 1263, 1264, 1429	
1235, 1267, 1357, 1392, 1432, 1433				<code>nohypertypes</code> (option)	<u>24</u>
<code>\newwrite</code>	39, 92,			<code>\noindent</code>	
189, 193, 199, 203, 485, 512,				330–334, 413–415, 599, 702–	
566, 569, 573, 576, 735, 762,				704, 714, 715, 851, 956–958,	
818, 821, 825, 828, 990, 1018,				968, 969, 1446–1449, 1451, 1452	
1072, 1075, 1080, 1083, 1144,				<code>\noist</code>	193, 195, <u>198</u> ,
1173, 1228, 1231, 1235, 1239,				199, 573, 671, 672, 825, 924,	
1300, 1330, 1385, 1388, 1392, 1395				925, 1075, 1076, 1079, 1080,	
<code>\nfss@text</code> 8, 469, 719, 973, 1128, 1283				1231, 1232, 1235, 1388, 1389, 1392	
				<code>nolangwarn</code> (option)	<u>25</u>
				<code>nolist</code> (option)	<u>14</u>
				<code>nolong</code> (option)	<u>13</u>
				<code>nomain</code> (option)	<u>21</u>
				<code>nonumberlist</code> (key)	<u>86</u>

debug	7	style-options	12
description	35	subentrycounter	16
disablemakegloss	37	symbols	44
dua	35	toc	10
entrycounter	16	translate	33
esclocations	13	ucmark	15
footnote	35	upmendex	36
hyperfirst	34	writeglslabelnames	39
index	45	writeglslabels	39
indexonlyfirst	34	xindy	37
kernelglossredefs	43	xindygloss	37
languages	34	xindynoglsnumbers	37
locales	34	\OR	628, 881
makeindex	36	\or	7, 10, 11, 33, 40, 41, 44, 45, 253, 412, 469, 471, 483, 487, 596, 677, 713, 718– 721, 732, 737, 847, 848, 930, 967, 973–975, 987, 991–994, 1103, 1127–1130, 1141, 1145, 1147, 1148, 1259, 1282, 1284, 1297, 1301, 1303, 1304, 1424, 1450
mendex	36	order (option)	36
mfirsttuc	45	\org@glossaryentrynumbers	226, 255, 587, 597, 839, 849, 1095, 1104, 1250, 1260, 1407, 1425
noglossaryindex	45	\org@glossarytitle	225, 226, 587, 838, 839, 1094, 1095, 1250, 1407
nogroupskip	15	\org@glspostdescription	54, 492, 742, 997, 1152, 1309
nohypertypes	24	\org@ifKV@glslink@hyper	132, 133, 537, 538, 788, 789, 1042, 1197, 1354
nolangwarn	25	\outputpenalty	320, 697, 951
nolist	14		
nolong	13		
nomain	21		
nonumberlist	13		
nopostdot	15		
noredefwarn	26		
nostyles	14		
nosuper	14		
notranslate	33		
notree	14		
nowarn	25		
numberedsection	11		
numberline	10		
numbers	44		
order	36		
preprocess-sort	30		
restoremakegloss	38		
sanitize	32		
sanitizesort	31		
saveglossarygroups	40		
savenumberlist	13		
savewrites	42		
section	10		
seeautonumberlist	13		
seenoindex	9		
shortcuts	35		
smallcaps	35		
smaller	35		
sort	17		
style	12		

578–580, 591, 592, 595, 596,	308, 329, 331–334, 380–382,
600, 601, 606, 607, 618, 619,	402–404, 406, 408, 410, 411,
623, 626, 640, 673, 720, 725,	413–419, 492, 599, 600, 674,
736, 740, 746, 750, 752, 755,	677–679, 691, 701–705, 712–
758–760, 765–771, 777, 787,	717, 742, 851, 852, 927, 931,
811, 825, 828, 830–832, 843,	932, 944, 945, 955–958, 966–
844, 847, 848, 852, 853, 858,	971, 997, 1152, 1309, 1446–1454
859, 871, 872, 876, 879, 893,	parent (key) <u>86</u>
926, 974, 979, 984, 990, 995,	\parindent 329–334, 403,
1002, 1006, 1008, 1011, 1013,	405, 406, 408, 410, 411, 413–
1015, 1016, 1021–1027, 1034,	415, 417–419, 677–679, 701–
1040, 1065, 1079, 1080, 1083,	705, 713–717, 930–932, 955–
1085, 1087, 1098–1100, 1103,	958, 966–968, 970, 971, 1446–1454
1104, 1107, 1108, 1110, 1113,	\parskip 329–
1129, 1134, 1139, 1145, 1150,	332, 404, 405, 411, 413, 414,
1156, 1161–1163, 1166, 1168,	701–703, 713–715, 955–957,
1170, 1171, 1176–1179, 1181,	966–968, 1446–1448, 1450–1452
1182, 1189, 1195, 1221, 1235,	\PassOptionsToPackage
1236, 1239, 1241, 1242, 1254, 421, 428, 631, 683, 884, 936
1255, 1258, 1259, 1263, 1265,	\pdfbookmark 350
1268, 1269, 1283, 1288, 1294,	\pdfstringdefPreHook
1300, 1307, 1313, 1318–1320, 216, 1090, 1245, 1402
1323, 1325, 1327, 1328, 1333–	\penalty 320, 697, 951
1336, 1338, 1339, 1346, 1352,	\PGLS 427, 687, 940
1378, 1392, 1393, 1396, 1398,	\Pgls 425, 425, 686, 939
1399, 1413, 1415–1417, 1420,	\pgls 424, 425, 685, 938
1424, 1425, 1428, 1429, 1431, 1434	\PGLSpl 427, 687, 940
\PackageInfo . . . 7, 8, 38, 200, 213,	\Pglspl 426, 426, 686, 939
469, 485, 573, 581, 718, 719,	\pglspl 424, 426, 685, 938
734, 735, 825, 833, 973, 989,	\phantomsection
1080, 1088, 1127, 1128, 1144,	. . . 60, 494, 744, 1000, 1154, 1311
1236, 1244, 1282, 1299, 1393, 1400	plural (key) <u>84</u>
\PackageWarning 7,	pluralaccess (key) <u>430</u>
8, 24, 46, 48, 469, 477, 718,	preprocess-sort (option) 30
719, 727, 972, 973, 981, 1127,	\prg 72, 76, 252
1128, 1136, 1282, 1291, 1304, 1305	\printacronyms
\PackageWarningNoLine <u>22</u> , 476, 725, 980, 1135, 1289
. 7, 8, 25, 420, 469,	\PrintChanges
477, 478, 666, 683, 718, 719, 6, 468, 718, 972, 1127, 1281
727, 919, 936, 972, 973, 981,	\printglossaries
982, 1127, 1128, 1136, 1282, 1291	205, <u>225</u> , 577, 586, 829, 838,
\pagegoal 320, 697, 951	1084, 1094, 1240, 1249, 1397, 1406
\pagelistname <u>51</u> , 54, 313–	\printglossary
315, 317, 318, 324, 326, 338–	22, 25, 44, 45, 205, <u>224</u> , 225,
341, 345–348, 490, 491, 694–	255, 476, 478, 487, 488, 577,
696, 700, 701, 707–709, 711,	586, 596, 726, 728, 738, 829,
712, 740, 741, 947, 948, 950,	838, 848, 980, 982, 993, 1084,
953–955, 960–962, 964–966,	1093, 1094, 1104, 1135, 1137,
995, 997, 1150, 1152, 1307, 1309	1147, 1148, 1240, 1249, 1259,
\par 54, 306,	1289, 1291, 1303, 1397, 1406, 1425

<code>\printindex</code>	300, 437, 438, 465, 470, 471,
... 45, 488, 738, 993, 1148, 1303	496, 497, 499, 501, 520, 522,
<code>\printnoidxglossaries</code>	523, 529–531, 535, 538, 539,
206, 225, 578, 586, 830, 838,	564, 581–583, 596, 598, 599,
1085, 1094, 1240, 1250, 1397, 1407	601, 605, 624, 628, 632, 637,
<code>\printnoidxglossary</code> 206, 209, 210,	638, 668, 669, 684, 688, 719,
225, 225, 239–241, 255, 577,	721, 724, 726, 735, 746, 748,
578, 580, 586, 591, 592, 596,	750, 751, 753, 755–757, 767–
829, 830, 832, 838, 843, 844,	771, 773, 774, 780–782, 786,
848, 1084, 1085, 1087, 1094,	788–790, 815, 816, 829, 833–
1098, 1099, 1104, 1240, 1242,	835, 838, 840, 841, 848–851,
1250, 1254, 1255, 1259, 1397,	853, 857, 877, 881, 885, 890,
1399, 1406, 1407, 1415, 1416, 1425	891, 914, 921, 922, 937, 941,
<code>\printnumbers</code>	974, 975, 978–980, 990, 1002,
... 45, 488, 738, 993, 1148, 1303	1003, 1005, 1007, 1008, 1011–
<code>\printsymbols</code>	1013, 1023–1027, 1029, 1030,
... 44, 487, 738, 993, 1147, 1303	1040, 1042, 1043, 1066, 1070,
<code>\ProcessOptions</code>	1071, 1082, 1084, 1085, 1088,
.... 421, 428, 631, 683, 884, 936	1090, 1091, 1093, 1096, 1097,
<code>\ProcessOptionsX</code>	1103, 1104, 1107, 1109, 1112,
... 46, 488, 738, 994, 1149, 1304	1122, 1125, 1128, 1130, 1133,
<code>\prop</code> 229–231, 1409, 1410	1135, 1144, 1157, 1158, 1160–
<code>\protect</code> 61, 127, 176,	1163, 1165–1167, 1177–1181,
277–279, 285, 301, 446, 447,	1184, 1185, 1194, 1195, 1197,
456, 457, 462, 495, 535, 607,	1198, 1221, 1225–1227, 1238–
608, 613, 617, 620, 622, 647,	1240, 1244, 1246, 1247, 1249,
653, 654, 658, 745, 785, 786,	1252, 1259, 1260, 1263, 1264,
859, 860, 865, 869, 873, 875,	1268, 1278, 1283, 1284, 1290,
900, 906, 907, 911, 1001, 1040,	1300, 1314, 1315, 1317–1320,
1065, 1114, 1115, 1119, 1120,	1322–1324, 1334–1338, 1341,
1125, 1155, 1194, 1195, 1221,	1342, 1351, 1352, 1354, 1355,
1269–1271, 1275, 1312, 1351,	1378, 1382–1384, 1395–1397,
1352, 1378, 1435, 1436, 1440, 1441	1401–1404, 1406, 1413, 1414,
<code>\protected@csedef</code>	1424, 1425, 1428, 1430, 1433, 1443
.. 99, 516, 767, 1023, 1178, 1335	<code>\protected@write</code>
<code>\protected@csxappto</code> 1125	... 10, 11, 79, 81, 143, 189–
<code>\protected@csxdef</code> 99,	191, 203, 204, 206, 209, 213,
516, 767, 1023, 1125, 1177, 1334	223, 263, 264, 302, 505, 566,
<code>\protected@eappto</code>	567, 576, 577, 580, 581, 586,
..... 39, 83, 109, 215, 735,	669, 688, 755, 756, 818, 819,
756, 757, 774, 834, 990, 1012,	828, 829, 832, 833, 838, 922,
1013, 1030, 1089, 1144, 1167,	941, 1011, 1012, 1073, 1074,
1184, 1245, 1300, 1324, 1341, 1402	1083, 1085, 1087, 1088, 1093,
<code>\protected@edef</code>	1125, 1165, 1166, 1206, 1228,
.... 9, 11, 23, 39, 64, 65, 68,	1229, 1239, 1240, 1243, 1244,
70, 71, 75, 79, 81–83, 99–105,	1249, 1322, 1323, 1363, 1385,
109, 110, 127, 132–134, 176,	1386, 1395–1397, 1399, 1401, 1406
184, 186, 202, 205, 206, 213,	<code>\protected@xdef</code> 23,
216, 219, 220, 223, 236, 237,	28, 90, 105, 108, 110, 111, 219,
254, 256, 264, 267, 275, 291,	226, 227, 433, 434, 474–476,

1189, 1190, 1195–1205, 1221,	1110, 1112–1121, 1126–1132,
1222, 1230–1235, 1238–1242,	1134–1137, 1139–1142, 1144–
1244, 1246–1248, 1250, 1251,	1149, 1151–1153, 1161, 1167–
1256, 1258, 1259, 1262, 1263,	1169, 1180, 1186, 1189, 1190,
1278, 1281, 1284, 1285, 1289,	1226–1228, 1232, 1233, 1238–
1296, 1297, 1300–1303, 1305,	1242, 1250, 1251, 1259, 1265,
1306, 1315, 1322, 1325, 1326,	1267–1277, 1281–1286, 1288–
1329, 1330, 1338–1340, 1344,	1291, 1295–1297, 1299, 1301–
1346, 1347, 1352–1362, 1378,	1304, 1306, 1308–1310, 1318,
1379, 1387–1392, 1395–1397,	1324–1326, 1337, 1343, 1346,
1399, 1401, 1403–1405, 1407,	1347, 1383–1385, 1389, 1390,
1408, 1420, 1421, 1423, 1424,	1395–1399, 1407, 1408, 1411,
1427, 1428, 1444, 1446, 1448–1454	1424, 1431, 1433–1442, 1446–1454
<code>\renewacronymstyle</code>	<code>\renewenvironment</code> 257, 303, 307,
. 277, 456–459, 461–464, 606,	310–313, 315, 316, 318–326,
653–656, 658–660, 858, 906–	328–348, 402, 411, 413, 414,
909, 911, 913, 1113, 1269, 1434	416, 598, 689, 691–695, 697–
<code>\renewcommand</code> 5–11, 13, 14,	716, 850, 942, 944, 946–968,
16, 17, 20–22, 24–26, 31, 33,	970, 1105, 1261, 1426, 1446–1453
34, 36, 38, 40–42, 44, 45, 49,	<code>\renewglossarystyle</code>
53–56, 70, 83, 86, 87, 102, 113,	. 265, 601, 853, 1107, 1263, 1428
117, 118, 184, 186, 188, 189,	<code>\RequireGlossariesLang</code>
195, 196, 202–208, 226, 228, 52, 53, 420, 490, 491,
234, 253, 254, 270, 274, 275,	666, 683, 740, 741, 919, 936,
277–287, 304, 305, 307–315,	996, 997, 1150, 1151, 1307, 1308
317, 318, 320, 322–326, 328–	<code>\RequirePackage</code>
339, 342–347, 411–415, 417– 4–6, 13, 14, 32, 33, 47,
419, 429, 432, 433, 441, 444,	52, 54, 292, 310, 316, 321, 327,
446, 447, 454–465, 468–473,	334, 342, 419–421, 429, 468,
475–478, 481–488, 490–492,	472, 482, 488, 490, 629, 631,
500, 506, 508, 518, 526, 527,	632, 666, 683, 692, 696, 698,
564–566, 570, 571, 576–579,	701, 705, 709, 718, 721, 722,
587, 588, 596, 605–616, 618,	731, 732, 738, 740, 882, 884,
619, 621–623, 625–627, 629,	885, 919, 936, 946, 949, 952,
632, 634, 640, 643, 645–647,	955, 958, 962, 971, 972, 976,
652–664, 667, 668, 672–682,	986, 987, 994, 996, 1122, 1126,
689–707, 709–711, 713–723,	1127, 1130, 1131, 1141, 1149,
725–728, 730–734, 736, 737,	1151, 1278, 1280, 1281, 1285,
739, 741, 742, 750, 757–759,	1296, 1304, 1307, 1309, 1443, 1446
769, 777, 778, 815–817, 822,	<code>\restorecounters@</code>
823, 828–831, 839, 840, 847,	. 133, 538, 789, 1043, 1198, 1355
848, 857–872, 874, 876, 878–	<code>restoremakegloss</code> (option) 38
880, 882, 885, 887, 888, 893,	<code>\romannumeral</code> 215, 218, 352–
896, 898, 900, 905–917, 920,	358, 365, 408, 409, 417, 418,
921, 925–935, 942–961, 963–	582, 583, 679, 716, 717, 834,
977, 979–982, 985–987, 989,	835, 932, 969, 970, 1089, 1090,
991, 993, 994, 996–998, 1006,	1245, 1246, 1402, 1403, 1453, 1454
1013–1015, 1025, 1032, 1034,	
1035, 1070–1072, 1077, 1078,	
1083–1086, 1095, 1096, 1103,	
	S
	<code>\s@glshyp@opt</code> 129,

129 , 536 , 787 , 1041 , 1195 , 1352	<code>\SetDescriptionAcronymDisplayStyle</code>
<code>\s@GlsSetXdyFirstLetterAfterDigits</code> 290 , 620 , 621 , 872 , 874
..... 188 ,	<code>\SetDescriptionAcronymStyle</code> ...
565 , 817 , 1071 , 1072 , 1227 , 1384 290 , 621 , 627 , 874 , 881
<code>\s@GlsSetXdyNumberGroupOrder</code> ..	<code>\SetDescriptionDUAAcronymDisplayStyle</code>
..... 188 , 189 , 565 , 566 , 290 , 618–620 , 871 , 872
817 , 1072 , 1227 , 1228 , 1384 , 1385	<code>\SetDescriptionDUAAcronymStyle</code>
<code>\s@ifglossaryexists</code> 71 , 72 , 290 , 619 , 627 , 872 , 881
224 , 501 , 586 , 751 , 752 , 838 ,	<code>\SetDescriptionFootnoteAcronymDisplayStyle</code>
1007 , 1094 , 1162 , 1249 , 1319 , 1406 290 , 616 , 618 , 869–871
<code>\s@newglossary</code>	<code>\SetDescriptionFootnoteAcronymStyle</code>
80 , 80 , 505 , 755 , 1011 , 1166 , 1323 290 , 618 , 627 , 870 , 881
<code>sanitize</code> (option) 32	<code>\SetDUADisplayStyle</code>
<code>sanitizesort</code> (option) 31 291 , 626 , 627 , 879 , 880
<code>\savecounters@</code>	<code>\SetDUASStyle</code> .. 291 , 627 , 628 , 880 , 881
. 133 , 538 , 789 , 1043 , 1197 , 1354	<code>\setentrycounter</code> .. 63 , 191 , 250 ,
<code>saveglossarygroups</code> (option) 40	264 , 496 , 567 , 595 , 600 , 667 ,
<code>savenumberlist</code> (option) 13	746 , 819 , 847 , 852 , 920 , 1002 ,
<code>savewrites</code> (option) 42	1074 , 1102 , 1107 , 1157 , 1229 ,
<code>sc-short-long</code> (acrstyle) 280	1258 , 1263 , 1314 , 1386 , 1422 , 1428
<code>sc-short-long-desc</code> (acrstyle) 282 , 458	<code>\SetFootnoteAcronymDisplayStyle</code>
<code>section</code> (option) 10 290 , 622 , 623 , 874 , 876
<code>see</code> (key) 85	<code>\SetFootnoteAcronymStyle</code>
<code>seeautonumberlist</code> (option) 13 291 , 623 , 628 , 876 , 881
<code>\seename</code> 52 , 222 , 490 , 585 , 740 , 837 ,	<code>\SetGenericNewAcronym</code>
996 , 1093 , 1150 , 1248 , 1307 , 1405	274 , 276 , 605 , 606 , 857 , 858 ,
<code>seenoindex</code> (option) 9	1112 , 1113 , 1267 , 1269 , 1433 , 1434
<code>\seq</code> 210 , 211 , 229–231 ,	<code>\setglossarypreamble</code>
234 , 251 , 252 , 300–302 , 1408–1411	... 57 , 493 , 743 , 998 , 1153 , 1310
<code>\setabbreviationstyle</code>	<code>\setglossarysection</code>
..... 292 , 880 , 1122 , 1278 , 1443	... 59 , 494 , 744 , 999 , 1154 , 1311
<code>\SetAcronymLists</code>	<code>\setglossarystyle</code>
... 24 , 477 , 726 , 981 , 1135 , 1290	. 225 , 253 , 264 , 293 , 307–309 ,
<code>\SetAcronymStyle</code> 292 ,	311–319 , 322–326 , 328–334 ,
483 , 484 , 627 , 628 , 733 , 880 , 881	336–341 , 343–345 , 347 , 412–
<code>\setacronymstyle</code> 270 , 276 , 292 , 606 ,	415 , 418 , 419 , 587 , 600 , 601 ,
858 , 880 , 1110 , 1113 , 1122 ,	629 , 691–697 , 699–715 , 717 ,
1265 , 1268 , 1278 , 1431 , 1434 , 1443	839 , 852 , 853 , 882 , 944–969 ,
<code>\setbool</code> 31 , 481 , 731 , 985 , 1140 , 1295	971 , 1094 , 1103 , 1107 , 1123 ,
<code>\setbox</code> 320 , 697 , 698 , 951	1250 , 1258 , 1263 , 1278 , 1407 ,
<code>\setcounter</code> ... 112 , 256 , 597 , 849 ,	1424 , 1428 , 1444 , 1446–1452 , 1454
1031 , 1104 , 1186 , 1260 , 1343 , 1425	<code>\setglossentrycompatibility</code> ...
<code>\SetCustomDisplayStyle</code>	260 , 595 , 599 , 601 , 847 , 851 , 853
..... 292 , 628 , 629 , 881 , 882	<code>\setkeys</code> 32 , 37 , 49 , 59 , 104 ,
<code>\SetCustomStyle</code> 292 , 629 , 882	132 , 186 , 226 , 275 , 481 , 484 ,
<code>\SetDefaultAcronymDisplayStyle</code>	489 , 494 , 519 , 537 , 564 , 565 ,
..... 289 , 615 ,	587 , 605 , 616 , 618 , 619 , 621 ,
616 , 867 , 868 , 1122 , 1278 , 1443	623 , 625 , 627 , 629 , 731 , 734 ,
<code>\SetDefaultAcronymStyle</code>	739 , 744 , 770 , 788 , 816 , 839 ,
..... 289 , 616 , 627 , 868 , 880	857 , 868 , 870 , 872 , 874 , 876 ,

878, 880, 882, 986, 989, 995, 999, 1026, 1042, 1071, 1094, 1112, 1140, 1143, 1149, 1154, 1181, 1197, 1226, 1227, 1250, 1267, 1295, 1299, 1306, 1311, 1338, 1354, 1383, 1384, 1407, 1433	\showglofield
	. 297, 631, 884, 1124, 1280, 1445
\setlength	\showglofirst
. 309, 310, 321, 329–332, 335, 342, 411, 413, 414, 418, 679, 692, 698, 701–703, 705, 709, 713–717, 932, 945, 946, 952, 955–959, 962, 963, 966–968, 970, 1446–1448, 1450–1452, 1454	. 293, 629, 883, 1123, 1278, 1444
\SetSmallAcronymDisplayStyle . .	\showglofirstaccess . . . 466, 665, 918
. 291, 624–626, 877–879	\showglofirsttpl
\SetSmallAcronymStyle 294, 629, 883, 1123, 1279, 1444
. 291, 625, 628, 878, 881	\showglofirstpluralaccess
\setstylefile 466, 665, 919
54, 492, 742, 997, 998, 1152, 1309	\showgloflag
\settodepth 113, 1032, 1187, 1344	. 297, 630, 884, 1124, 1280, 1445
\settoheight	\showgloindex
. 113, 545, 797, 1032, 1187, 1344	. 297, 630, 884, 1124, 1280, 1445
\settowidth 113, 278, 373–377, 608, 678, 679, 716, 717, 860, 932, 969, 970, 1032, 1115, 1187, 1270, 1344, 1436	\showglolevel
\setupglossaries 293, 629, 882, 1123, 1278, 1444
. . . 49, 488, 739, 994, 1149, 1306	\showgloloclist
\sfcode . 15, 472, 722, 976, 1131, 1285	. 297, 631, 884, 1124, 1280, 1445
short (key) 88	\showglolong
short-long (acrstyle) 279, 456	. 297, 630, 884, 1124, 1279, 1445
short-long-desc (acrstyle) . . 281, 458	\showglolongaccess 467, 666, 919
shortaccess (key) 431	\showglolongplural 297
shortcuts (option) 35	\showglolongpluralaccess
shortplural (key) 88 467, 666, 919
shortpluralaccess (key) 431	\showgloname
\show . 293–299, 466, 467, 629–631, 665, 666, 882–884, 918, 919, 1123, 1124, 1278–1280, 1444–1446	. 295, 630, 883, 1123, 1279, 1444
\showacronymlists	\showglonameaccess 466, 665, 918
. 298, 631, 884, 1124, 1280, 1445	\showgloparent
\showglocounter 293, 629, 882, 1123, 1278, 1444
. 294, 629, 883, 1123, 1279, 1444	\showgloplural
\showglodesc 293, 629, 882, 1123, 1278, 1444
. 295, 630, 883, 1123, 1279, 1444	\showglopluralaccess . . 466, 665, 918
\showglodescaccess 466, 666, 919	\showgloshort
\showglodescplural 296, 630, 883, 1124, 1279, 1445
. 296, 630, 883, 1124, 1279, 1445	\showgloshortaccess . . . 467, 666, 919
\showglodescpluralaccess	\showgloshortplural 296
. 466, 666, 919	\showgloshortpluralaccess
 467, 666, 919
	\showglosort
	. 296, 630, 883, 1124, 1279, 1445
	\showglossaries
	. 298, 631, 884, 1124, 1280, 1445
	\showglossarycounter
	. 298, 631, 884, 1124, 1280, 1445
	\showglossaryentries
	. 299, 631, 884, 1124, 1280, 1446
	\showglossaryin
	. 298, 631, 884, 1124, 1280, 1445
	\showglossaryout
	. 298, 631, 884, 1124, 1280, 1445

<code>\showglossarytitle</code>	624, 646, 647, 653–661, 667–
. 298 , 631 , 884 , 1124 , 1280 , 1445	671, 673–681, 689–693, 698,
<code>\showglosymbol</code>	705, 709, 713–717, 720, 734–
. 296 , 630 , 883 , 1124 , 1279 , 1445	737, 740, 746, 749, 751, 758–
<code>\showglosymbolaccess</code> ..	466 , 665 , 919
<code>\showglosymbolplural</code>	760, 777, 779, 785–787, 815,
. 296 , 630 , 883 , 1124 , 1279 , 1445	818–821, 825–832, 837, 839,
<code>\showglosymbolpluralaccess</code>	840, 849, 853, 855, 859–867,
..... 466 , 665 , 919	873, 877, 900, 906–914, 920–
<code>\showglotext</code>	924, 927, 929–935, 942–946,
. 293 , 629 , 882 , 1123 , 1278 , 1444	952, 959, 963, 966–970, 974,
<code>\showglotextaccess</code>	466 , 665 , 918
<code>\showglotype</code>	989, 990, 992, 995, 1002, 1004–
. 294 , 629 , 883 , 1123 , 1279 , 1444	1007, 1011, 1014–1016, 1034,
<code>\showglouseri</code>	1036, 1037, 1040, 1070, 1073–
. 294 , 630 , 883 , 1123 , 1279 , 1444	1075, 1079–1085, 1087, 1089,
<code>\showglouserii</code>	1093, 1094, 1096, 1105, 1110,
. 294 , 630 , 883 , 1123 , 1279 , 1444	1114–1122, 1126, 1129, 1143–
<code>\showglouseriii</code>	1145, 1147, 1150, 1156, 1159–
. 294 , 630 , 883 , 1123 , 1279 , 1444	1161, 1165, 1169, 1170, 1189,
<code>\showglouseriv</code>	1191, 1194, 1195, 1225, 1228–
. 295 , 630 , 883 , 1123 , 1279 , 1444	1231, 1235–1242, 1245, 1249–
<code>\showglouserv</code>	1251, 1260, 1261, 1266, 1269–
. 295 , 630 , 883 , 1123 , 1279 , 1444	1278, 1283, 1299, 1300, 1302,
<code>\showglouservi</code>	1307, 1313, 1316–1318, 1322,
. 295 , 630 , 883 , 1123 , 1279 , 1444	1326, 1327, 1346, 1348, 1351,
<code>\skip</code>	402 , 404 , 405 , 410
<code>sm-short-long</code> (acrstyle)	280
<code>sm-short-long-desc</code> (acrstyle)	282 , 459
<code>smallcaps</code> (option)	35
<code>smaller</code> (option)	35
<code>\SmallNewAcronymDef</code>	291 ,
465 , 624 , 626 , 664 , 877 , 879 , 917	
<code>sort</code> (key)	84
<code>sort</code> (option)	17
<code>\space</code>	9, 10,
38 , 40 , 43 , 44 , 50 , 63 , 67 , 68 ,	
70 , 79 , 86 , 88 , 117 , 119 – 121 ,	
126 , 127 , 184 , 190 – 193 , 198 ,	
200 – 206 , 209 , 215 , 223 , 225 ,	
228 , 255 , 257 , 271 , 277 , 279 –	
287 , 291 , 303 , 305 , 307 , 308 ,	
311 , 322 , 335 , 342 , 343 , 411 –	
415 , 417 , 418 , 446 , 447 , 456 –	
465 , 470 , 484 – 487 , 490 , 496 ,	
499 , 501 , 508 , 509 , 526 , 528 ,	
529 , 535 , 536 , 563 , 566 – 569 ,	
573 – 580 , 585 , 587 , 588 , 597 ,	
598 , 601 , 604 , 607 – 615 , 620 ,	
624 , 646 , 647 , 653 – 661 , 667 –	
671 , 673 – 681 , 689 – 693 , 698 ,	
705 , 709 , 713 – 717 , 720 , 734 –	
737 , 740 , 746 , 749 , 751 , 758 –	
760 , 777 , 779 , 785 – 787 , 815 ,	
818 – 821 , 825 – 832 , 837 , 839 ,	
840 , 849 , 853 , 855 , 859 – 867 ,	
873 , 877 , 900 , 906 – 914 , 920 –	
924 , 927 , 929 – 935 , 942 – 946 ,	
952 , 959 , 963 , 966 – 970 , 974 ,	
989 , 990 , 992 , 995 , 1002 , 1004 –	
1007 , 1011 , 1014 – 1016 , 1034 ,	
1036 , 1037 , 1040 , 1070 , 1073 –	
1075 , 1079 – 1085 , 1087 , 1089 ,	
1093 , 1094 , 1096 , 1105 , 1110 ,	
1114 – 1122 , 1126 , 1129 , 1143 –	
1145 , 1147 , 1150 , 1156 , 1159 –	
1161 , 1165 , 1169 , 1170 , 1189 ,	
1191 , 1194 , 1195 , 1225 , 1228 –	
1231 , 1235 – 1242 , 1245 , 1249 –	
1251 , 1260 , 1261 , 1266 , 1269 –	
1278 , 1283 , 1299 , 1300 , 1302 ,	
1307 , 1313 , 1316 – 1318 , 1322 ,	
1326 , 1327 , 1346 , 1348 , 1351 ,	
1352 , 1382 , 1385 – 1388 , 1392 –	
1397 , 1399 , 1402 , 1406 – 1408 ,	
1426 , 1431 , 1435 – 1443 , 1450 – 1454	
<code>\spacefactor</code>	15, 472, 722, 976, 1131, 1285
<code>\SS</code>	29, 480, 730, 984, 1139, 1294
<code>\ss</code>	29, 480, 730, 984, 1139, 1294
<code>\str</code> ..	251, 263, 1107, 1262, 1423, 1428
<code>\string</code>	9–11, 20, 21, 25,
30, 38, 40, 43, 44, 50, 61–68,	
70, 79, 81, 86, 88, 94, 97, 98,	
109, 110, 117, 119–121, 134,	
135, 137–139, 141, 143, 184,	
187–195, 198, 199, 203–207,	
209, 215, 220, 221, 225, 228,	
239–241, 255, 264, 270, 292,	
302, 467, 468, 470, 475, 478,	
484–487, 490, 495–501, 505,	
508, 509, 512, 513, 515, 523,	
526, 528, 529, 535, 536, 539,	
541, 542, 544, 563, 565–570,	
572, 573, 576–580, 584, 585,	
587, 588, 591, 592, 596, 599–	
601, 667–672, 688, 720, 725,	
728, 734–737, 740, 745, 746,	
748–751, 755, 756, 758–760,	

763–766, 774, 777–779, 786,	
787, 790, 792–794, 796, 815–	
822, 824, 825, 828–832, 836,	
837, 839, 840, 843, 844, 848,	
851, 853, 880, 920–925, 941,	
974, 979, 982, 984, 989, 990,	
992, 995, 1001–1007, 1011,	
1012, 1014–1016, 1019–1022,	
1030, 1034–1037, 1043, 1044,	
1046, 1047, 1049, 1050, 1070–	
1077, 1079, 1080, 1083–1085,	
1087, 1089, 1091, 1092, 1094–	
1096, 1098, 1099, 1104, 1110,	
1122, 1125, 1129, 1134, 1137,	
1139, 1143–1145, 1147, 1150,	
1155–1161, 1165, 1166, 1169,	
1170, 1174, 1176, 1177, 1184,	
1185, 1189–1191, 1198–1202,	
1204, 1206, 1225, 1227–1232,	
1234–1236, 1239–1243, 1245,	
1247, 1248, 1250, 1251, 1254,	
1255, 1259, 1265, 1278, 1283,	
1288, 1291, 1294, 1299, 1300,	
1302, 1307, 1312–1318, 1322,	
1323, 1326, 1327, 1331, 1333,	
1334, 1341, 1342, 1346–1348,	
1355–1359, 1361, 1363, 1382,	
1384–1389, 1391–1393, 1395–	
1400, 1402, 1404, 1405, 1407,	
1408, 1415, 1416, 1425, 1431, 1443	
\strut	243, 307–309, 311, 312, 314,
	322, 323, 325, 335, 337, 339,
	343, 344, 346, 392, 406, 415,
	600, 673–676, 678, 680–682,
	691–694, 698–700, 705–707,
	709–711, 715, 852, 927–929,
	931, 933–935, 944–948, 952–
	954, 959–961, 963–965, 969, 1452
style (option)	12
style-options (option)	12
styles:	
altlist	308
altlistgroup	308
altlisthypergroup	309
altlong4col	315
altlong4col-booktabs	318
altlong4colborder	316
altlong4colheader	315
altlong4colheaderborder	316
altlongragged4col	325
altlongragged4col-booktabs	319
altlongragged4colborder	326
altlongragged4colheader	325
altlongragged4colheaderborder	326
altsuper4col	340
altsuper4colborder	341
altsuper4colheader	340
altsuper4colheaderborder	341
altsuperragged4col	346
altsuperragged4colborder	347
altsuperragged4colheader	347
altsuperragged4colheaderborder	347
alttree	416
alttreegroup	418
alttreehypergroup	419
index	411
indexgroup	412
indexhypergroup	412
inline	303
list	307
listdotted	309
listgroup	307
listhypergroup	307
long	310
long-booktabs	317
long3col	312
long3col-booktabs	317
long3colborder	313
long3colheader	313
long3colheaderborder	313
long4col	313
long4col-booktabs	318
long4colborder	314
long4colheader	314
long4colheaderborder	315
longborder	311
longheader	311
longheaderborder	311
longragged	321
longragged-booktabs	319
longragged3col	323
longragged3col-booktabs	319
longragged3colborder	324
longragged3colheader	324
longragged3colheaderborder	324
longraggedborder	322
longraggedheader	322
longraggedheaderborder	323

mcolalttree	333	335, 337, 339, 343, 344, 346,
mcolalttreegroup	333	402, 411, 413, 415, 417, 523,
mcolalttreehypergroup	333	587, 593, 594, 599, 600, 690–
mcolalttreespannav	334	694, 698–700, 705–707, 709–
mcolindex	329	711, 713–716, 774, 839, 845,
mcolindexgroup	329	851, 943–948, 952–954, 959–
mcolindexhypergroup	329	961, 963–965, 967, 968, 970,
mcolindexspannav	330	1030, 1095, 1101, 1106, 1185,
mcoltree	330	1250, 1256, 1257, 1262, 1342,
mcoltree*	327	1407, 1418, 1421, 1427, 1450–1453
mcoltreegroup	331	\subitem 329, 410–412, 677, 701, 712,
mcoltreehypergroup	331	713, 930, 955, 966, 967, 1446, 1450
mcoltreenoname	331	sublistdotted (style) 309
mcoltreenonamegroup	332	\subsubitem 329, 411, 412, 677, 701,
mcoltreenonamehypergroup	332	713, 930, 955, 966, 967, 1446, 1450
mcoltreenonamespannav	332	super (style) 335
mcoltreespannav	331	super3col (style) 337
sublistdotted	309	super3colborder (style) 337
super	335	super3colheader (style) 338
super3col	337	super3colheaderborder (style) 338
super3colborder	337	super4col (style) 338
super3colheader	338	super4colborder (style) 339
super3colheaderborder	338	super4colheader (style) 339
super4col	338	super4colheaderborder (style) 340
super4colborder	339	superborder (style) 336
super4colheader	339	superheader (style) 336
super4colheaderborder	340	superheaderborder (style) 336
superborder	336	superragged (style) 342
superheader	336	superragged3col (style) 344
superheaderborder	336	superragged3colborder (style) 345
superragged	342	superragged3colheader (style) 345
superragged3col	344	superraggedborder (style) 343
superragged3colborder	345	superraggedheader (style) 343
superragged3colheader	345	superraggedheaderborder (style) 344
superraggedborder	343	superraggedright3colheaderborder
superraggedheader	343	(style) 345
superraggedheaderborder	344	symbol (key) 85
superraggedright3colheaderborder	345	symbolaccess (key) 430
tree	413	\symbolname 51, 54, 314, 315,
tree*	348	318, 326, 339–341, 347, 348,
treegroup	413	490, 491, 695, 696, 701, 707–
treehypergroup	414	709, 712, 740, 741, 948, 950,
treenoname	414	954, 955, 961, 962, 965, 966,
treenonamegroup	415	995, 997, 1150, 1152, 1307, 1309
treenonamehypergroup	415	symbolplural (key) 85
subentrycounter (option)	16	symbolpluralaccess (key) 430
\subglossentry	110, 226,	symbols (option) 44
	244, 248, 261, 304, 307–309,	\sys 47, 48, 1304, 1305
	311, 312, 314, 322, 323, 325,	

T

<code>\t</code>	29, 479, 729, 983, 1138, 1293	657, 658, 661–665, 667, 668,
<code>\tablehead</code>	335–348, 705–712, 959–966	671, 740, 741, 746, 750, 751,
<code>\tabletail</code>	335–348, 705–712, 959–966	755, 791–797, 818, 821–824,
<code>\tabularnewline</code>	311–315,	832, 833, 835, 838, 845, 850,
	317, 318, 322–326, 335–348,	851, 854, 857–860, 863, 865,
	675, 676, 680, 681, 692–696,	867–871, 873, 875, 877–879,
	698–701, 705–712, 929, 933,	881, 882, 885, 906, 907, 910,
	934, 946–950, 952–955, 959–966	911, 914–918, 920, 922, 924,
<code>\texorpdfstring</code>	175, 216, 561, 813,	995, 996, 1002, 1005, 1007,
	1065, 1090, 1221, 1245, 1378, 1402	1011, 1045–1050, 1072, 1075,
<code>\text</code>	4, 5, 233,	1077, 1079, 1087, 1088, 1090,
	234, 971, 972, 1126, 1281, 1411	1093, 1101, 1109, 1112–1115,
<code>text</code> (key)	84	1118, 1119, 1150, 1151, 1156,
<code>textaccess</code> (key)	430	1160, 1161, 1166, 1200–1205,
<code>\textbar</code>	303, 689, 942, 1126	1228, 1231, 1233–1235, 1243,
<code>\textbf</code>	268, 360, 361, 410, 599, 603,	1244, 1246, 1249, 1256, 1264,
	676–679, 712, 851, 855, 930–	1268–1270, 1274, 1275, 1307,
	932, 966, 1109, 1265, 1430, 1449	1308, 1313, 1317, 1318, 1323,
<code>\textit</code>	268, 603, 855, 1109, 1265, 1430	1357–1362, 1385, 1388, 1390–
<code>\textmd</code>	268, 603, 855, 1109, 1265, 1430	1392, 1400, 1401, 1403, 1406,
<code>\textrm</code>	268, 603, 854, 1109, 1265, 1430	1421, 1430, 1433–1436, 1439, 1440
<code>\textsc</code>	268, 279, 280, 286,	<code>\the@numberlist</code>
	463, 603, 609, 614, 618, 621,	184, 185,
	623, 626, 660, 855, 860, 861,	564, 815, 816, 1070, 1226, 1383
	866, 871, 874, 876, 879, 913,	<code>\theglossary</code>
	1109, 1115, 1116, 1120, 1265,	. . 6, 7, 468, 718, 972, 1127, 1281
	1271, 1276, 1430, 1436, 1437, 1442	<code>theglossary</code> (env.)
<code>\textsf</code>	268, 603, 855, 1109, 1265, 1430	257
<code>\textsl</code>	268, 603, 855, 1109, 1265, 1430	<code>\theglossaryentry</code>
<code>\textsmaller</code>	280,	. 15, 257, 473, 597, 722, 849,
	286, 464, 609, 614, 618, 622,	977, 1105, 1131, 1260, 1286, 1426
	623, 626, 660, 861, 866, 871,	<code>\theglossarysubentry</code>
	874, 876, 879, 913, 1115, 1116,	. 16, 257, 473, 598, 723, 849,
	1121, 1271, 1272, 1276, 1437, 1442	977, 1105, 1132, 1261, 1286, 1426
<code>\texttt</code>	268, 603, 855, 1109, 1265, 1430	<code>\theHglentrycounter</code>
<code>\textulc</code>	270, 603, 855, 1110, 1266, 1431	133,
<code>\textup</code>	268, 270, 603, 855,	134, 216, 219, 538, 582, 583,
	1109, 1110, 1265, 1266, 1430, 1431	668, 789, 834, 835, 921, 1042,
<code>\TH</code>	29, 480, 730, 984, 1139, 1294	1043, 1090, 1091, 1197, 1198,
<code>\th</code>	29, 480, 730, 984, 1139, 1294	1245, 1246, 1354, 1355, 1402, 1403
<code>\the</code>	50, 53, 63, 68, 70, 81, 137–142,	<code>\theH</code> . . 221, 584, 836, 1092, 1248, 1404
	189, 193, 196, 198, 212, 213,	<code>\theHglossaryentry</code>
	218, 223, 248, 267, 275–279,	. . . 15, 473, 722, 977, 1131, 1286
	283, 285, 456, 460, 462, 465,	<code>\theHglossarysubentry</code>
	489, 491, 496, 499, 501, 505,	. . . 16, 473, 723, 977, 1132, 1286
	540–545, 566, 569, 571, 572,	<code>\theHglentrycounter</code> 133, 134, 216,
	580, 581, 583, 586, 593, 598,	219, 538, 539, 582, 583, 789,
	599, 602, 605–608, 612, 613,	834, 835, 1043, 1090, 1091,
	615–626, 628, 632, 653, 654,	1198, 1245, 1246, 1355, 1402, 1403
		<code>\thepage</code>
		143, 1206, 1363
		<code>\thesection</code>
		. . . 50, 489, 739, 995, 1150, 1307

<code>\this@dialect</code>	.. 53, 420, 491, 666, 683, 741, 919, 936, 997, 1151, 1308
<code>\tiny</code> 8, 470, 719, 973, 1128, 1283
<code>\tl</code> 5, 17–19, 30, 31, 47, 48, 58, 72, 76, 112, 143, 210, 228, 230, 231, 233–235, 242–246, 251, 252, 262, 327, 328, 352–360, 362–380, 388–390, 392–402, 404, 405, 972, 1126, 1186, 1206, 1281, 1287, 1288, 1305, 1343, 1363, 1408–1412, 1417–1419, 1428
<code>toc (option)</code> <u>10</u>
<code>\token</code> 210, 264
<code>\toks@</code> 50, 53, 63, 68, 70, 81, 136–142, 196, 198, 223, 267, 489–491, 496, 499, 501, 505, 540–545, 570–572, 586, 598, 599, 602, 632, 667, 740, 741, 746, 750, 751, 755, 791–797, 822–824, 838, 850, 851, 854, 885, 920, 995, 996, 1002, 1005, 1007, 1011, 1045–1050, 1077, 1079, 1093, 1108, 1109, 1150, 1151, 1156, 1160, 1161, 1166, 1200–1205, 1233–1235, 1249, 1264, 1307, 1308, 1313, 1317, 1318, 1323, 1357–1362, 1390–1392, 1406, 1430
<code>\toprule</code> 317, 318, 696, 949, 950
<code>\TrackIfKnownLanguage</code> 52, 996, 1151, 1308
<code>\trans@languages</code> 53, 490, 740, 996, 1151, 1308
<code>\translate</code>	. 53, 54, 491, 741, 742, 996, 997, 1151, 1152, 1308, 1309
<code>translate (option)</code> <u>33</u>
<code>\transletelet</code> 21, 22, 44, 45, 475, 476, 487, 488, 725, 726, 738, 979, 980, 993, 1134, 1135, 1148, 1289, 1303
<code>tree (style)</code> <u>413</u>
<code>tree* (style)</code> <u>348</u>
<code>treegroup (style)</code> <u>413</u>
<code>treehypergroup (style)</code> <u>414</u>
<code>treenoname (style)</code> <u>414</u>
<code>treenonamegroup (style)</code> <u>415</u>
<code>treenonamehypergroup (style)</code> <u>415</u>
<code>\triangleright</code> 8, 470, 719, 973, 1128, 1283
<code>\ttfamily</code>	8, 470, 719, 973, 1128, 1283
<code>\TX@trial</code> 113, 524, 775, 1032, 1187, 1344
<code>type (key)</code> <u>85</u>
<code>\typeout</code>	25, 477, 727, 981, 1136, 1291
U	
<code>\u</code> 29, 479, 729, 983, 1138, 1293
<code>\uccode</code>	247, 593, 845, 1100, 1256, 1420
<code>ucmark (option)</code> <u>15</u>
<code>\undef</code>	87, 93, 224, 508, 512, 586, 759, 763, 838, 1015, 1019, 1093, 1169, 1173, 1249, 1326, 1330, 1406
<code>\unexpanded</code> 45, 994, 1148, 1304
<code>\unskip</code>	.. 102, 309, 518, 674, 692, 769, 927, 945, 1025, 1180, 1337
<code>\unvbox</code> 320, 697, 698, 951
<code>upmendex (option)</code> <u>36</u>
<code>\use</code> 230, 236, 242, 244, 245, 251, 374–376, 382–385, 387–393, 399, 1413, 1417, 1418, 1423
<code>\usedictionary</code> 53, 490, 741, 996, 1151, 1308
<code>\usepackage</code>	30, 239–241, 270, 467, 468, 591, 592, 843, 844, 984, 1098, 1099, 1110, 1139, 1254, 1255, 1265, 1294, 1415, 1416, 1431
<code>user1 (key)</code> <u>87</u>
<code>user1access (key)</code> <u>431</u>
<code>user2 (key)</code> <u>87</u>
<code>user2access (key)</code> <u>431</u>
<code>user3 (key)</code> <u>87</u>
<code>user3access (key)</code> <u>431</u>
<code>user4 (key)</code> <u>88</u>
<code>user4access (key)</code> <u>431</u>
<code>user5 (key)</code> <u>88</u>
<code>user5access (key)</code> <u>432</u>
<code>user6 (key)</code> <u>88</u>
<code>user6access (key)</code> <u>432</u>
<code>\UTFviii@two@octets</code> 30, 480, 730, 984, 1139, 1294
<code>\UTFviii@two@octets@combine</code> 30, 480, 730, 984, 1139, 1294
V	
<code>\v</code> 29, 480, 729, 984, 1138, 1293
<code>\value</code> 112, 1031, 1186, 1343
<code>\vbox</code> 320, 697, 698, 951
<code>\vsize</code> 320, 321, 698, 951

<code>\vskip</code>	306,	1232, 1235, 1385, 1388, 1389, 1392
	320, 329, 410, 691, 697, 701,	
	712, 944, 951, 955, 966, 1446, 1449	
<code>\vss</code>	320, 697, 698, 951	
W		
<code>\warn@nomakeglossaries</code>	202 , 204 ,	
	205 , 207 , 576–578 , 827–830 ,	
	1082–1085 , 1238–1241 , 1395–1397	
<code>\warn@noprntglossary</code>	204–	
	206 , 223 , 227 , 576–578 , 586 ,	
	587 , 828–830 , 838 , 839 , 1083–	
	1085 , 1093 , 1095 , 1239 , 1240 ,	
	1249 , 1251 , 1396 , 1397 , 1406 , 1408	
<code>\write</code>	39 , 94 , 119 , 189–	
	194 , 206 , 212 , 228 , 485 , 512 ,	
	513 , 528 , 566–570 , 574 , 575 ,	
	578 , 581 , 588 , 668–672 , 735 ,	
	763 , 778 , 779 , 818–822 , 826 ,	
	827 , 830 , 832 , 840 , 922–925 ,	
	990 , 1019 , 1035 , 1036 , 1072–	
	1076 , 1080–1083 , 1085 , 1088 ,	
	1095 , 1096 , 1145 , 1174 , 1190 ,	
	1228–1232 , 1236–1238 , 1240 ,	
	1243 , 1251 , 1300 , 1305 , 1331 ,	
	1347 , 1385–1389 , 1397 , 1400 , 1408	
<code>writetgslabelnames</code> (option)	39	
<code>writetgslabels</code> (option)	39	
<code>\writeist</code>	189 , 199 , 566 , 569 , 570 ,	
	573 , 668 , 671 , 672 , 818 , 820 ,	
	822 , 825 , 922 , 924 , 925 , 1072 ,	
	1075 , 1076 , 1080 , 1228 , 1231 ,	
X		
<code>\x</code>	602	
<code>\xatlevel@</code>		
	. 133 , 538 , 789 , 1042 , 1197 , 1354	
<code>\xcapitalisewords</code>	561	
<code>\xdef</code>	106 , 520 , 522 , 587 , 688 ,	
	771 , 941 , 942 , 1027 , 1182 , 1339	
<code>\xglsaccsupp</code> . .	438 , 438 , 637 , 638 , 891	
<code>\xglsfieldaccsupp</code>	437 , 637 , 890	
<code>\xifinlist</code> . .	92 , 762 , 1018 , 1173 , 1330	
<code>\xifinlistcs</code> . .	236–238 , 241 , 589 ,	
	592 , 840 , 841 , 844 , 1096 , 1097 ,	
	1099 , 1252 , 1255 , 1413 , 1414 , 1416	
<code>xindy</code> (option)	37	
<code>xindygloss</code> (option)	37	
<code>xindynoglsnumbers</code> (option)	37	
<code>\xmakefirstuc</code>		
 529–531 , 535 , 559 , 560 ,	
	684 , 780–782 , 786 , 811 , 812 , 937	
<code>\xspace</code>	269 ,	
	603 , 855 , 1109 , 1265 , 1430 , 1431	
Y		
<code>\year</code>	189 , 193 , 566 , 569 ,	
	668 , 671 , 818 , 821 , 922 , 924 ,	
	1072 , 1075 , 1228 , 1231 , 1385 , 1388	
Z		
<code>\Z</code>	262 , 1106 , 1262 , 1427	
<code>\z@</code>	320 , 697 , 698 , 951	